

UNIVERSIDADE FEDERAL DO PARANÁ

**AUTOMATIZAÇÃO DE UM TESTE DE ASSOCIAÇÃO DE PALAVRAS PARA
RECONHECIMENTO DE PADRÕES DE PAPÉIS SUGERIDOS**

Curitiba, 2018

UNIVERSIDADE FEDERAL DO PARANÁ

Cleverson Zocche Sato

Luis Henrique de Souza

Marcelo Marzola Leite

Renata Soares Pereira

**AUTOMATIZAÇÃO DE UM TESTE DE ASSOCIAÇÃO DE PALAVRAS PARA
RECONHECIMENTO DE PADRÕES DE PAPÉIS SUGERIDOS**

Trabalho de conclusão de curso apresentado como requisito parcial à obtenção do grau tecnológico em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientador: Prof. Dr. Alexander Robert Kutzke

Curitiba, 2018

AGRADECIMENTOS

Somos agradecidos a algumas pessoas importantes que fizeram parte de nossa trajetória na graduação. Nossos familiares por todo apoio e paciência nos momentos mais difíceis. Nossos amigos e colegas de curso por momentos de estudo e também momentos de descontração.

Agradecemos aos nossos professores pela dedicação em passar o conhecimento e nos orientar. Em especial agradecemos ao nosso orientador deste trabalho de conclusão de curso, professor Dr. Alexander Robert Kutzke, por ter acreditado em nós e simpatizado com nosso projeto desde o início.

Enfim, somos gratos a todos que de alguma forma nos deram apoio nessa fase de nossas vidas, com certeza foram muito importantes para a realização desse projeto.

“São as nossas escolhas... mais que as nossas capacidades que mostram
quem realmente somos”
(J. K. ROWLING)

RESUMO

Foi construído como trabalho de conclusão de curso uma versão automatizada do teste psicológico de associação de palavras com o foco em utilizar tecnologias modernas para aplicá-lo e fazer com que este identifique qual dentre três papéis foi inicialmente atribuído a um sujeito experimental. Para tanto foi feito o uso de conceitos de ludificação para tentar aumentar o engajamento do usuário ao teste, assemelhando-o em sua proposta a um jogo que o desafia a enganar a correção automatizada, utilizando uma *api* de reconhecimento de voz para capturar as respostas do sujeito e o tempo de reação aos estímulos realizados através de palavras pronunciadas pelo sistema. Estes dados são submetidos a uma rede neural artificial para que esta realize a sugestão de qual seria o papel do sujeito. Possui uma interface gráfica para permitir a análise de dados dos testes de forma ágil, imediata e comparativa. Com isso procura-se demonstrar a utilização de um sistema computacional como alternativa à resolução do problema da complexidade apresentado pelo teste psicológico e como este pode auxiliar na pesquisa científica e análise de seus resultados.

Palavras-chave: Teste de associação de palavras, inteligência artificial, automação, ludificação, reconhecimento de fala, análise de dados.

ABSTRACT

An automated version of the psychological test of word association was constructed as an end-of-course work focusing on using modern technologies in its application and making it identify which out of three roles was initially assigned to then experimental subject. In order to do so, it were used gamification concepts to try to increase the user's engagement to the test, resembling it to a game in its proposition when challenging him to deceive the automated correction, used a voice recognition api to capture the answers and the time of reaction of the subject to the stimuli of words pronounced by the system. These data are submitted to an artificial neural network so that it suggests which would be the subject's role. It has a graphical interface to allow the analysis of test data in an fast, immediate and comparative way. The aim of this work is to demonstrate the use of a computational system as an alternative to solve the complex problem presented by this psychological test and how it can aid in scientific research and the analysis of its results.

Key-words: Word Association test, artificial intelligence, automatization, gamification, speech recognition, data analysis

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Neurônio biológico..... | 24 |
| Figura 2 - Neurônio artificial..... | 25 |
| Figura 3 - Rede Neural Artificial MLP..... | 26 |
| Figura 4 - Scrum..... | 31 |
| Figura 5 - Cronograma..... | 32 |
| Figura 6 - Arquitetura do sistema..... | 46 |
| Figura 7 - Módulos do sistema..... | 47 |
| Figura 8 - Fluxo de telas..... | 48 |
| Figura 9 - Caso de uso geral..... | 59 |
| Figura 10 - Caso de uso Teste..... | 60 |
| Figura 11 - Caso de uso RNA..... | 61 |
| Figura 12 - Casos de uso da análise de dados..... | 62 |
| Figura 13 - Tela Inicial..... | 63 |
| Figura 14 - Tela de Consentimento..... | 64 |
| Figura 15 - Tela de dados..... | 65 |
| Figura 16 - Tela de preparação..... | 66 |
| Figura 17 - Tela de orientações 1..... | 67 |
| Figura 18 - Tela de orientações 2..... | 68 |
| Figura 19 - Tela de calibragem..... | 69 |
| Figura 20 - Tela de atribuição de papel..... | 70 |
| Figura 21 - Tela de teste..... | 71 |
| Figura 22 - Tela de resultado bom..... | 72 |
| Figura 23 - Tela de ruim para o personagem Ladrão..... | 73 |
| Figura 24 - Tela de ruim para o personagem Testemunha..... | 73 |
| Figura 25 - Tela de ruim para o personagem Espião..... | 74 |
| Figura 26 - Tela de Login..... | 75 |
| Figura 27 - Tela de Análise de dados de um teste..... | 76 |
| Figura 28 - Tela de Análise de todos os teste..... | 76 |
| Figura 29 - Tela de Análise da rede neural..... | 77 |
| Figura 30 - Sequência instruções..... | 78 |
| Figura 31 - Sequência Dados..... | 79 |
| Figura 32 - Sequência Calibragem..... | 80 |
| Figura 33 - Sequência Aplicação..... | 81 |
| Figura 34 - Sequência Resultado..... | 82 |
| Figura 35 - Sequência Análise de dados..... | 82 |

| | |
|--|----|
| Figura 36 - Diagrama de classes..... | 83 |
| Figura 37 - Diagrama físico de banco de dados..... | 84 |

LISTA DE ABREVIATURAS E SIGLAS

- API - Interface de programação de aplicação
- DER - Diagrama entidade relacionamento
- GUI - Interface gráfica do usuário
- HTML - Linguagem de marcação de hipertexto
- p. - Página
- SQL - Linguagem de consulta estruturada
- TCC - Trabalho de conclusão de curso
- UML - Linguagem de Modelagem Unificada
- XML - Extensible Markup Language

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO..... | 12 |
| 1.1 PROBLEMA..... | 12 |
| 1.2 JUSTIFICATIVA..... | 13 |
| 1.3 OBJETIVO..... | 14 |
| 1.3.1 <i>Objetivos gerais</i> | 14 |
| 1.3.2 <i>Objetivos Específicos</i> | 14 |
| 1.4 ESTRUTURA DO DOCUMENTO..... | 15 |
| 2 DESENVOLVIMENTO..... | 16 |
| 2.1 FUNDAMENTAÇÃO TEÓRICA..... | 16 |
| 2.1.1 <i>O teste de associação de palavras</i> | 16 |
| 2.1.1.1 <i>Critérios de Correção</i> | 17 |
| 2.1.2 <i>Ludificação</i> | 19 |
| 2.1.3 <i>Reconhecimento de palavras</i> | 20 |
| 2.1.4 <i>Correção automatizada</i> | 20 |
| 2.1.4.1 <i>Agentes inteligentes</i> | 21 |
| 2.1.4.2 <i>Aprendizagem Supervisionada</i> | 22 |
| 2.1.4.3 <i>Redes Neurais Artificiais</i> | 23 |
| 2.1.4.4 <i>Neurônio</i> | 23 |
| 2.1.4.5 <i>Perceptron Multicamadas</i> | 25 |
| 2.1.4.6 <i>Algoritmo Backpropagation</i> | 26 |
| 2.2 METODOLOGIA..... | 28 |
| 2.2.1 <i>Diferenças do teste original com o proposto</i> | 28 |
| 2.2.2 <i>Cronograma e Planejamento</i> | 29 |
| 2.2.2.1 <i>Sprints 1 e 2</i> | 34 |
| 2.2.2.2 <i>Sprint 3</i> | 34 |
| 2.2.2.3 <i>Sprint 4</i> | 34 |
| 2.2.2.4 <i>Sprints 5 e 6</i> | 34 |
| 2.2.2.5 <i>Sprint 7</i> | 35 |
| 2.2.2.6 <i>Sprint 8</i> | 35 |
| 2.2.2.7 <i>Sprint 9</i> | 35 |
| 2.2.2.8 <i>Sprint 10</i> | 35 |
| 2.2.2.9 <i>Sprint 11</i> | 36 |

| | |
|---|-----------|
| 2.2.2.10 Sprint 12..... | 36 |
| 2.2.2.11 Sprint 13..... | 36 |
| 2.2.2.12 Sprint 14..... | 36 |
| 2.2.2.13 Sprint 15..... | 36 |
| 2.2.3 Tecnologias para a construção do sistema..... | 36 |
| 2.2.4 Vídeos de papéis..... | 38 |
| 2.2.5 Speech Recognition..... | 38 |
| 2.2.6 Implementação das Redes Neurais..... | 40 |
| 2.2.7 Análise de dados..... | 43 |
| 2.2.8 Modelagem..... | 44 |
| 2.2.9 Arquitetura do sistema..... | 44 |
| 2.2.10 Definição do sistema..... | 46 |
| 2.3 RESULTADOS..... | 49 |
| 3 CONSIDERAÇÕES FINAIS..... | 51 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | 52 |
| GLOSSÁRIO..... | 55 |
| APÊNDICE A - PAPÉIS..... | 56 |
| APÊNDICE B - PALAVRAS..... | 58 |
| APÊNDICE C - CASOS DE USO..... | 59 |
| APÊNDICE D - DIAGRAMAS DE SEQUÊNCIA..... | 78 |
| APÊNDICE E - DIAGRAMA DE CLASSES..... | 83 |
| APÊNDICE F - DIAGRAMA DE BANCO DE DADOS..... | 84 |

1 INTRODUÇÃO

O teste psicológico de Associação de Palavras é consideravelmente especial por ter uma relação íntima com a própria história da psicologia científica e da avaliação psicológica. Inicialmente concebido em 1887/1888 por Francis Galton, logo em seguida foi utilizado por outros pesquisadores, que desenvolveram diversas variações deste. (MERTEN, 1992)

Com mais de cem anos de existência, o teste que possibilitou vários avanços e pesquisas no campo da psicologia foi escolhido ser atualizado mais uma vez, desta vez utilizando-se dos recursos modernos que possuímos na atualidade.

Para isto, foi desenvolvido um sistema web para a aplicação do teste, de maneira que o usuário possa participar do teste sem que o aplicador esteja presente. O software, denominado Detector, conduz o participante por meio de instruções de como o teste funcionará. Orienta também a verificação dos seus periféricos de entrada e saída de áudio.

Após esta etapa, atribui-se aleatoriamente ao usuário o personagem de um jogo, entre três possibilidades: espião, testemunho ou criminoso. O personagem é contextualizado a partir de um vídeo, que situa o teste na estória.

Os dados coletados durante o processo de aplicação do teste são tratados e transmitidos para uma rede neural. Após este processo, é esperado que a rede neural, treinada, determine qual o personagem foi sorteado no começo da aplicação.

Todas as informações ficam disponíveis para consulta do aplicador, em ambiente com acesso restrito. Gráficos são utilizados para facilitar a análise, evidenciando as relações entre as aplicações do teste.

1.1 PROBLEMA

Apesar de sua importância histórica, o teste de associação de palavras é consideravelmente difícil de ser aplicado e corrigido, tanto por consumir

uma quantidade considerável de tempo como por necessitar de um trabalho concentrado e preciso do aplicador na marcação de tempo e anotação das respostas, algo que aumenta consideravelmente a possibilidade de erros (JUNG, 2012, p. 250).

E além dos problemas técnicos, a análise de dados resultante desta é gigantesca e complexa (idem, p. 17), uma vez que cada aplicação pode resultar em 200 novas respostas ou mais, cada uma com uma frase, tempo de reação e categorização, sendo ainda muito difícil estabelecer correlações e fazer uma análise efetiva dos dados.

Todos esses fatores contribuíram para que este teste fosse aos poucos sendo substituído por metodologias diferentes de avaliação psicológica, por mais que ainda possa contribuir para esta área e para pesquisas científicas de modo geral.

1.2 JUSTIFICATIVA

Diante dessas características pouco atrativas, este teste na realidade se torna perfeito para um trabalho de atualização. Convenientemente, todas as dificuldades para a realização deste podem ser melhoradas ou mesmo totalmente contornadas pelo uso de tecnologias computacionais modernas, as quais permitem a automatização da aplicação do teste, uso de ferramentas de *data mining* para a análise dos dados e até mesmo a realização de uma correção automatizada deste.

Uma importante informação que justifica o motivo da criação de um sistema baseado em um teste de associação de palavras é referente a formação acadêmica e profissão de um dos integrantes do grupo do presente trabalho de conclusão de curso, psicólogo formado pela Universidade Federal do Paraná este contribuiu abundantemente na concepção da ideia de automatizar esse teste de forma a transformá-lo em um “jogo” com algumas características do teste original.

E é com o objetivo de aplicar essas novas tecnologias à este procedimento clássico da psicologia e possivelmente ajudar a atingir conclusões que antes eram pouco visíveis aos pesquisadores, que este

trabalho de conclusão de curso é proposto e tem sua relevância.

1.3 OBJETIVO

1.3.1 Objetivos gerais

Construir uma versão moderna e automatizada do teste de associação de palavras.

1.3.2 Objetivos Específicos

- Pesquisar sobre tecnologias modernas de interesse para a aplicação do teste de associação de palavras
- Contornar ou minimizar os problemas do teste de associação de palavras com o uso das tecnologias pesquisadas
- Construir um teste autônomo que consiga resultados similares ou levante variáveis comportamentais/psicológicas similares ao da pesquisa original
- Engajar o sujeito experimental a responder a pesquisa seriamente apesar de seu caráter virtual, utilizando dos conceitos de ludificação
- Tornar fácil a aplicação e análise de dados do teste ao automatizá-lo
- Elaborar uma rede neural artificial para fazer a correção automatizada do teste
- Utilizar de sistemas informatizados de obtenção e análise de dados que permitam uma visualização gráfica e estatística deste
- Utilizar tecnologias de reconhecimento de voz para a captura das respostas do sujeito experimental
- Tentar obter novas variáveis e informações significativas sobre o comportamento humano através do teste

1.4 ESTRUTURA DO DOCUMENTO

Este documento está estruturado da seguinte maneira:

A parte de fundamentação teórica tem o intuito de apresentar, detalhadamente, os conceitos aplicados e comparando algumas tecnologias testadas para a implementação do sistema.

A parte de metodologia faz a apresentação dos materiais e métodos utilizados no projeto, explicando as tecnologias e também apresentando o cronograma com as atividades e suas respectivas atribuições aos membros da equipe.

A parte de resultados é composta pela apresentação do sistema, onde é demonstrada a sua arquitetura, o fluxo de telas, suas descrições, funcionalidades e resultados obtidos.

As considerações finais concluem este projeto mostrando os pontos positivos e negativos do mesmo, possíveis melhorias e planos futuros.

Ao fim deste documento, estão as referências bibliográficas, os apêndices contendo os diagramas de casos de uso com as imagens das telas, diagramas de classes, diagramas do banco de dados e diagramas de sequência.

2 DESENVOLVIMENTO

Por este trabalho ser baseado em conhecimentos consideravelmente específicos e técnicos, foi necessário realizar pesquisas bibliográficas para fundamentar a construção desse sistema.

2.1 FUNDAMENTAÇÃO TEÓRICA

Neste subcapítulo são apresentados conceitos importantes para o desenvolvimento desse projeto. É essencial deixar clara a teoria para que se possa colocar em prática o uso de ferramentas e até mesmo a aplicação e construção de um novo software.

2.1.1 *O teste de associação de palavras*

O teste de Associação de palavras no qual este trabalho foi baseado tem sua metodologia fundamentada na versão utilizada por Carl Gustav Jung (2012). Este foi utilizado pelo pesquisador no início do século XX, dando os fundamentos para sua teoria dos complexos, sendo posteriormente estudado seu uso na criminologia (MERTEN, 1992). Por não ser o foco deste trabalho, será omitida a teoria psicológica por trás desta, focando-se apenas no que for importante para sua aplicação.

Com relação à aplicação, esta se dava da seguinte forma: O aplicador dizia à pessoa experimental uma série de palavras-estímulo, pedindo-lhe que respondesse o mais rápido possível com a primeira palavra que viesse à mente. O tempo entre a palavra-estímulo e a reação é medido e, após aproximadamente 100 palavras-estímulo, estas eram repetidas à pessoa experimental, solicitando que esta repetisse exatamente as respostas que deu a cada palavra-estímulo anteriormente. Terminado o procedimento, o pesquisador pedia para que a pessoa experimental explicasse suas associações e então as respostas eram avaliadas de acordo com seus tipos, tempo de reação e se sua repetição foi correta ou não. (JUNG, 2012, p.353)

Todo este procedimento tem como objetivo identificar o que Jung (2012, p. 654 a 661) chamou de “complexos autônomos”, que seriam temas específicos que despertam reações emocionais da pessoa experimental e, independentemente da vontade desta, se manifestavam psicofisicamente quando palavras de significados relacionados eram utilizadas no teste.

Exemplificando com uma pessoa que esteja envolvida em algum tipo de crime, esta normalmente responde de uma forma padronizada às palavras estímulo, mas quando há a “justiça” é utilizada, a pessoa tende a ter uma reação emocional involuntária e não consciente, que interfere no padrão até então estabelecido, causando um tempo de resposta diferente ou reações defensivas (como “falhas” de fala ou audição, repetições, uso de palavras estereotipadas ou maior dificuldade em lembrar, na segunda passagem, o que ela havia respondido) que podem interferir inclusive em respostas posteriores.

A identificação das palavras que causam uma ativação de um complexo resulta na possibilidade de se inferir sobre o conteúdo deste e como a pessoa, emocionalmente, se encontra no momento da aplicação do teste, sendo que já foram realizadas pesquisas com diversos grupos, como epiléticos, esquizofrênicos e até suspeitos de crimes com resultados promissores. (JUNG, 2012)

2.1.1.1 Critérios de Correção

Ao se considerar os resultados do teste de associação de palavras, eram levados em conta principalmente o tempo de reação às palavras-estímulo, o sucesso ou não da repetição da resposta e a categoria em que esta se enquadra.

Estas variáveis são relativamente auto-explicativas, à exceção da categorização das palavras. Estas eram divididas nos seguintes grupos principais: (idem, p. 22-52)

a) Associação interna (de “semelhança”)

- Coordenação: conteúdos semelhantes, como lago - mar.

- Relação predicativa: julgamentos ou qualidades, como lago - bonito; rezar - pessoa piedosa.

- Dependência causal: relação de causa - efeito, como cortar - doloroso.

b) Associação externa (de “continuidade”)

- Coexistência: vínculo de existência no tempo/espço. Aluno - professor; mesa - cadeira.

- Identidade: sinônimos. Magnífico - esplêndido; bom - ótimo.

- Forma linguístico-motora relações treinadas/habituais. Vovô - ivo viu a uva, doce - amargo, associação - de mulheres, assassino - assassinar.

c) Reação de som

- Complementação de palavra: junção para criar nova palavra. Paca -embu; Mato-Grosso; Milagre-iro.

- Som: sons similares. Mala - Bala; carro - jarro; vela - velha.

- Rima: Valente - potente; final - animal.

d) Grupo Residual (outros)

- Reação indireta: relação necessita de um elo intermediário. Planície - (nevada) - branco, andar - (sob as) - árvores)

- Reação sem sentido: nem palavras, nem associações. Pai - frt...grsd.

- Falha: ausência de reação.

- Palavra-estímulo repetida: a palavra estímulo é repetida antes da resposta. Casa - casa vermelha.

e) Outras classificações de interesse:

- Perseveração: a resposta é influenciada por uma palavra-estímulo anterior. Inverno - patins > lago - gelo ; tampa - caixa > ratos - cesto.

- Reação egocêntrica: respostas fortemente referenciadas ao eu ou com juízos subjetivos. Pai - sou; calcular - eu; não sei; piano - horrendo.

- Repetição: repetição de uma palavra ou de um padrão. Bala -

mala > avião - mala.

- Conexão Linguística: relações gramaticais, de número de sílabas ou algumas sílabas iguais. BoneCO - COzido; correr - (verbo) - calcular.

Apesar de todas as categorias serem interessantes, como por exemplo o grande uso de respostas de “reação egocêntrica” ser um indicativo de uma personalidade narcisista, os fatores mais significativos de complexo, neste sentido, são as repetições de palavras já utilizadas antes, uso de grandes frases nas respostas, falhas na percepção da palavra-estímulo ou emissão da resposta e a perseveração. (idem)

Com estas informações, era possível ter uma certa percepção dos complexos emocionais da pessoa experimental. Por isso, em alguns casos, especialmente nos relacionados a investigações de crimes e eventos, Jung utilizava palavras-estímulo específicas com o que desejava descobrir, chamando-as de “palavras críticas”, as quais eram misturadas com palavras neutras, chamadas de “palavras neutras” ou “palavras pós-críticas” quando estas estavam logo após as palavras críticas. Nestes, era levando em consideração na avaliação se as reações emocionais ocorriam durante as palavras críticas ou as pós-críticas, de modo a definir se a pessoa experimental tinha algum complexo emocional relacionado ao que era de interesse do pesquisador. (p.505, 506)

Assim sendo, no que diz respeito à construção de um teste nesses moldes, são essas as variáveis principais que devem ser avaliadas.

2.1.2 Ludificação

A ludificação, no sentido derivado do conceito em inglês “*gamification*”, se refere a utilização de conceitos e características de jogos em contextos que não são jogos de forma a tentar obter o engajamento das pessoas (KIM, 2015). Apesar de ser um termo relativamente recente, ganhando destaque maior devido às novas possibilidades que as tecnologias móveis trouxeram, o

uso de ludificação como uma estratégia de marketing já existe desde a década de 60, nos Estados Unidos. (PRINCE, 2013)

De modo geral, esta se baseia na “teoria da diversão”, que diz que o método mais fácil para mudar o comportamento de uma pessoa é utilizando-se da diversão. (KIM, 2015)

A função deste conceito para este sistema será melhor descrita na seção de metodologia.

2.1.3 Reconhecimento de palavras

O reconhecimento de voz consiste em, dado um sinal acústico, reconhecer uma palavra, ou uma sequência delas, ditas por um falante (RUSSEL, NORVIG 2013).

Esta tecnologia consiste na transmissão das vibrações sonoras provocadas pela fala de uma pessoa para uma máquina através da transformação destas em sinais elétricos por microfone, onde estes sinais são analisados e convertidos pelo programa de reconhecimento de voz em estruturas fonéticas e então em palavras. Geralmente, o programa de reconhecimento de voz utiliza-se de informações do contexto (normalmente obtidas e tratadas utilizando alguma técnica de inteligência artificial) para otimizar seu funcionamento. (ZWASS, 2018)

É uma das principais aplicações de inteligência artificial, cada vez mais sistemas usam esse recurso, seja para fazer pesquisas na internet, tradução de textos entre outros. Uma ótima escolha quando o sistema precisa ser operado com as mãos livres ou quando precisa ser mais dinâmico.

2.1.4 Correção automatizada

Com intuito de automatizar uma tarefa, nesse projeto foram utilizadas ferramentas que pudessem fazer esse trabalho. Nos próximos subcapítulos são introduzidos os conceitos sobre inteligência artificial para que se possa

entender como foi construída a correção automatizada no presente sistema.

2.1.4.1 Agentes inteligentes

Nem sempre é possível prever todos os problemas que o sistema ou componente dele terá que resolver. Se assim fosse, então a aprendizagem de máquina se faria desnecessária. Pode-se melhorar muito as tarefas através da aprendizagem, como um programa que antecipa e se adapta às diferentes condições do mercado de ações, um algoritmo de reconhecimento facial ou até mesmo um robô capaz de transitar em labirintos diferentes. Para tanto são necessários quatro fatores: O *Componente* a ser melhorado, *Conhecimento disponível* do agente, que *Representação* será dada aos dados e ao componente e qual *Feedback* se tem para a aprendizagem de máquina definidos a seguir:

- Agente: Segundo Stuart Russel e Peter Norvig, 2013: “*um agente é algo que percebe e age em um ambiente*”. Ele estará sempre aprendendo conforme for observando coisas sobre o mundo.
- Componente: Um componente pode ser considerado como uma parte do que precisa ser aprendido pelo agente.
- Conhecimento disponível (prévio): É como o próprio nome diz, o conhecimento prévio que se tem para que o agente aprenda alguma tarefa.
- Representação: Forma como os dados e o componente será representado.
- *Feedback*: O feedback é o que determina qual tipo de aprendizagem de máquina será utilizada, como melhor explicado nos próximos parágrafos.

Os chamados *Componentes* podem ser aprendidos recebendo uma alimentação de dados necessária. Por exemplo, se o agente estiver aprendendo a dirigir um automóvel, sempre que seu instrutor lhe diga para

frear, o agente tem a possibilidade de aprender um componente. Ou quando o agente vê imagens de um caminhão ele pode aprender a reconhecer caminhões, um outro componente.

Os componentes do agente podem ser representados de diversas formas: usando Redes Bayesianas, Sentenças Lógicas, vetores de valores e atributos formando uma Representação Fatorada, Aprendizagem Indutiva usando pares de entrada e saída ou até Aprendizagem Analítica/Dedutiva que parte de uma regra conhecida para uma nova regra derivada e útil.

Os três tipos principais de aprendizagem de máquina são determinados pelos três tipos de feedbacks. A Aprendizagem Não Supervisionada onde o agente pode aprender padrões de entrada, sendo uma tarefa bem comum o Agrupamento. Na Aprendizagem por Reforço o agente é ensinado a partir de várias recompensas ou punições. Já na Aprendizagem Supervisionada o agente recebe alguns pares de entrada/saída como exemplo e aprende a mapear a entrada e saída. Como o presente trabalho tem enfoque no uso de Redes Neurais Artificiais, será explicada a Aprendizagem Supervisionada no próximo subcapítulo. (Russel, Norvig 2013, p. 606)

2.1.4.2 Aprendizagem Supervisionada

Na Aprendizagem Supervisionada se tem um conjunto de entradas e saídas, como se houvesse um “professor” para avaliar se a saída está de acordo com o esperado. Os pesos (pesos armazenam o conhecimento) vão sendo calculados e alterados fazendo com que seja alcançada a resposta esperada através dos erros encontrados. É uma aprendizagem a partir de exemplos.

São exemplos de uso de aprendizagem supervisionada: Reconhecimento facial e de outros objetos; Detecção de spam, Classificação de textos em categorias baseadas em características comuns e palavras chaves; e Previsão de preços futuros com base em cotações históricas.

Uma propriedade importante de uma rede neural artificial é o

aprendizado. A rede aprende a partir de iterações que ajustam os pesos, esse processo é feito por um algoritmo computacional. Em redes neurais, HAYKIN (1999) traz a definição de aprendizado: “Aprendizado é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estímulo pelo ambiente no qual a rede está inserida”

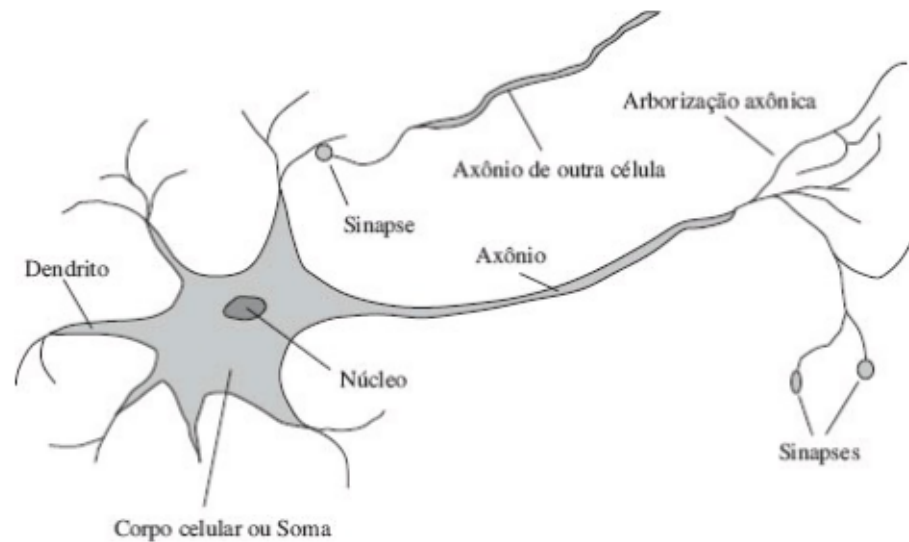
2.1.4.3 Redes Neurais Artificiais

As redes neurais artificiais são baseadas no principal componente do sistema nervoso, o neurônio. Todo ser multicelular possui algum tipo de sistema nervoso e este é responsável pela aprendizagem do organismo através de entrada de informação. Este sistema é capaz de modificar o comportamento do ser vivo através de experiências.

2.1.4.4 Neurônio

Como se sabe, o neurônio é fundamental quando se trata de redes neurais artificiais (HAYKIN, 1999). Um neurônio é formado por Corpo celular ou Soma, Núcleo celular, localizado na Soma, ramificações que partem do Corpo celular formando os Dendritos e uma ramificação bem mais longa chamada de *Axônio*, como ilustrado na Figura 1. Um neurônio é capaz de fazer conexões com 10-100.000 outros neurônios. Essas conexões são feitas a partir das *Sinapses*, sinais transmitidos de uma célula nervosa a outra com reações eletroquímicas. Esse processo de comunicação consiste em um neurônio pré-sináptico enviar um impulso nervoso através do axônio, este impulso chega até sua extremidade e libera neurotransmissores para o outro neurônio. São esses sinais que controlam a atividade do cérebro em curto e longo prazo e acredita-se que esses mecanismos formam a aprendizagem cerebral.

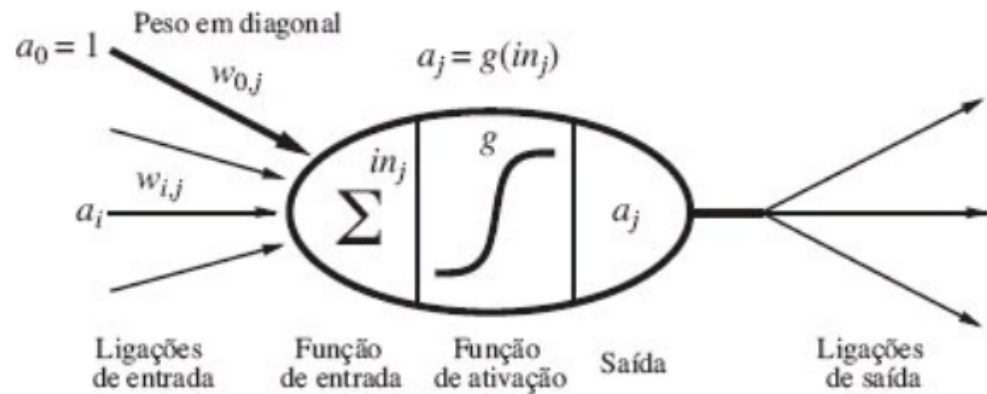
Figura 1 - Neurônio biológico



FONTE: Russel, Norvig (2013)

Assim, uma rede neural artificial é composta por unidades ou nós e em cada unidade há *Ligações de Entrada*, estas servem para receber o sinal do neurônio anterior; *Pesos*, que contém a informação, o conhecimento aprendido; *Função de entrada*, que faz o cálculo da soma ponderada das entradas; *Função de Ativação*, essa função é aplicada no resultado da soma da função de entrada; *Saída*, que será o resultado dos cálculos anteriores; e *Ligações de Saída*, que são ligações da unidade com outras próximas unidades. (Russel, Norvig 2013, p. 634, 635)

Figura 2 - Neurônio artificial



FONTE: Russel, Norvig (2013)

A figura 2 representa um neurônio artificial. Este foi o primeiro modelo de neurônio artificial e foi desenvolvido em 1943 por McCulloch & Pitts inspirado no que se sabia sobre um neurônio biológico. (Zambiasi, 2011)

2.1.4.5 Perceptron Multicamadas

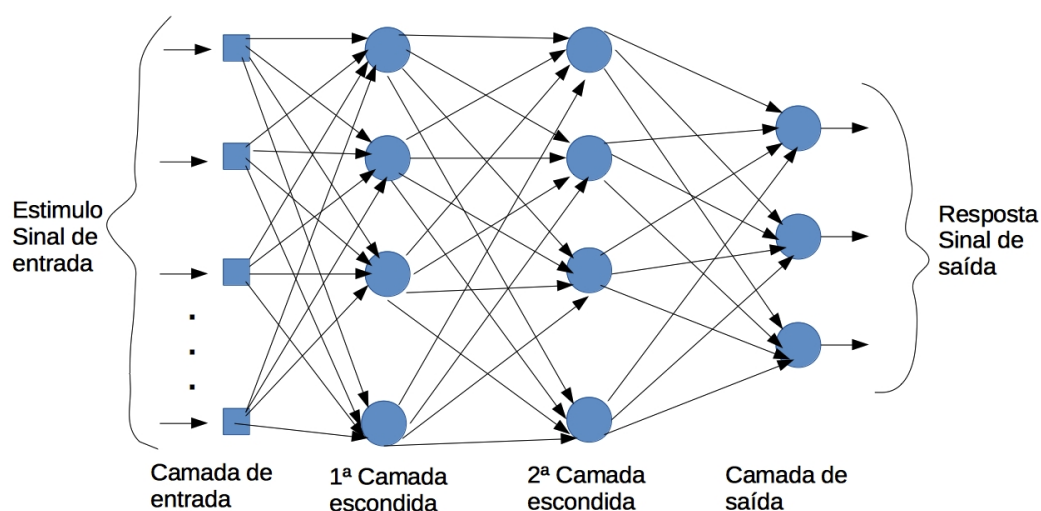
Perceptron Multicamadas (MLP - MultiLayer Perceptron) é uma das mais populares arquiteturas de redes neurais artificiais encontradas na literatura. Seja por ser flexível na formação de soluções para um grande número de problemas quanto por sua propriedade de aproximação universal. MLP tem sido aplicada com sucesso na solução de problemas diversos. (IYODA, 2000)

Uma rede Perceptron Multicamadas é composta por uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. Cada camada pode conter vários nós. O sinal é propagado através da rede, camada por camada, desde a camada de entrada a camada de saída.

Cada neurônio desta rede possui uma função de ativação não-linear, As funções de ativação basicamente tomam a decisão sobre se um neurônio deve ou não ser ativado. Em outras palavras, se a informação que o neurônio recebeu é relevante ou deve ser ignorada. (Deep Learning Book, 2018)

A Figura 3 representa um exemplo de uma rede com quatro nós de entrada, três de saída e duas camadas internas (escondidas).

Figura 3 - Rede Neural Artificial MLP



FONTE: Os Autores (2018).

Neste modelo de rede, os neurônios trocam informação entre si através das sinapses, desde a camada de entrada até a camada de saída.

2.1.4.6 Algoritmo Backpropagation

O algoritmo Backpropagation foi desenvolvido em meados de 1980 por Rumelhart, Hinton e Williams, é um dos mais populares algoritmos de aprendizado de redes neurais MLP.

Este algoritmo precisa receber junto com a entrada um padrão de saída. Assim, é feito o processamento camada a camada até que seja

produzida uma saída. Essa saída é comparada com o padrão de saída esperado. Se a saída resposta estiver errada, esse erro é calculado e retro-propagado desde a camada de saída até a primeira camada de entrada, atualizando os pesos das camadas internas. Essa iteração acontece até que o valor da saída seja o valor esperado.

A seguir apresenta-se um algoritmo geral de retropropagação (backpropagation) retirado do livro Inteligência Artificial de Russel, Norvig, 2013, p. 640 que exemplifica o funcionamento deste:

“

função APRENDIZAGEM-DE-RETRO-PROP(exemplos, rede) **retorna**
uma rede neural

entradas: *exemplos*, um conjunto de exemplos, cada um com um vetor de entrada **x** e vetor de saída **y**

rede, uma rede multicamadas com L camadas, pesos $w_{i,j}$ e função de ativação g

variáveis locais: Δ , um vetor de erros, indexado pelo nó de rede

repetir

para cada peso $w_{i,j}$ na rede **faça:**

$w_{i,j}$ = um número randômico pequeno

para cada exemplo (\mathbf{x}, \mathbf{y}) em *exemplos* **faça:**

/ Propagar as entradas para a frente para computar as saídas */*

para cada nó i c camada de entrada **faça:**

$a_i = x_i$

para $l = 2$ até L **faça**

$in_j = \sum_i w_{i,j} a_i$

$a_j = g(in_j)$

/ Propagar deltas retrocedendo da camada de saída para a camada de entrada */*

para cada nó j na camada de saída **faça**

$\Delta[j] = g'(in_j) \times (y_j - a_j)$

para cada $l = L - 1$ até 1 **faça**

para cada nó i na camada l **faça**

$\Delta[i] = g'(in_i) \sum_j w_{i,j} \Delta[j]$

```

/* Atualiza cada peso na rede usando deltas
*/
para cada peso  $w_{i,j}$  na rede faça
     $w_{i,j} = w_{i,j} + a \times a_i \times \Delta[j]$ 
até que algum critério de parada seja satisfeito
retorno rede
    ”

```

2.2 METODOLOGIA

A partir da teoria pesquisada, foi traçado um planejamento e a metodologia da construção deste trabalho.

Dado o interesse em utilizar tecnologias consideravelmente novas, optou-se por construir um sistema com caráter experimental e com algumas inovações não testadas ou não completamente consolidadas até o momento, assim como métodos diferentes dos canônicos para o teste em si. Enquanto deixar o projeto muito mais rico e permite a obtenção de novos conhecimentos, há uma chance consideravelmente maior de falhas no resultado final.

2.2.1 Diferenças do teste original com o proposto

No processo de implementação das novas tecnologias na construção deste teste, foi necessário fazer algumas mudanças no método de aplicação com relação ao teste original, dadas as características intrínsecas entre os computadores e as pessoas. Isso é um problema considerável, pois assim não há nenhuma garantia de que assim seja obtido o resultado esperado do teste, ou seja, que seja possível identificar os “complexos emocionais” ativados em cada resposta.

Provavelmente, a maior diferença é o fato de que o aplicador do teste não é um ser humano focado em analisar o sujeito experimental que está em uma sala especificamente designada para tanto, mas sim uma máquina imparcial que potencialmente pode aplicar este em qualquer lugar com uma conexão à internet. Nestas condições, há a hipótese de que as reações

emocionais de uma pessoa ao responder o teste automatizado, seu engajamento, seja muito reduzido.

Somando-se a isso, há o problema ético da realização de experiências psicológicas em seres humanos, algo não compatível com um sistema criado sem a avaliação prévia de um comitê de ética.

De forma a tentar contornar esses problemas, foi utilizado uma estratégia de ludificação, transformando este teste psicológico de uma avaliação pessoal em uma espécie de jogo impessoal, um desafio ao sujeito experimental para que este, tendo recebido um papel (um personagem de uma história) previamente, tente enganar nossa máquina, enquanto este tenta adivinhar qual o papel que a pessoa em questão está assumindo. Uma alteração significativa do original, mas da qual levanta-se a hipótese de que os mecanismos envolvidos podem ser os mesmos do que os do teste original.

Com relação à obtenção das variáveis, optou-se por utilizar aquelas em que o computador teria vantagem, que são as dependentes de medição de tempo e cálculo matemático, em detrimento das que dependem de semântica e observações subjetivas. Assim sendo, são extraídos de cada teste o tempo de reação para cada palavra apresentada e se a resposta foi adequada ou não ao solicitado, o que em teoria seria o bastante para a análise de dados.

E por último, no que diz respeito à correção do teste, é utilizado de uma técnica de inteligência artificial para tentar descobrir uma correlação objetiva de uma relação consideravelmente subjetiva entre tempo de reação e um estímulo verbal e oferecer uma sugestão de correção ao pesquisador. De modo a complementar essa correção e facilitar a extração de conhecimentos a partir desses dados da pesquisa e das redes neurais, são também utilizadas técnicas de análise de dados, o que talvez seja a maior vantagem do uso dessas técnicas modernas na aplicação experimental deste.

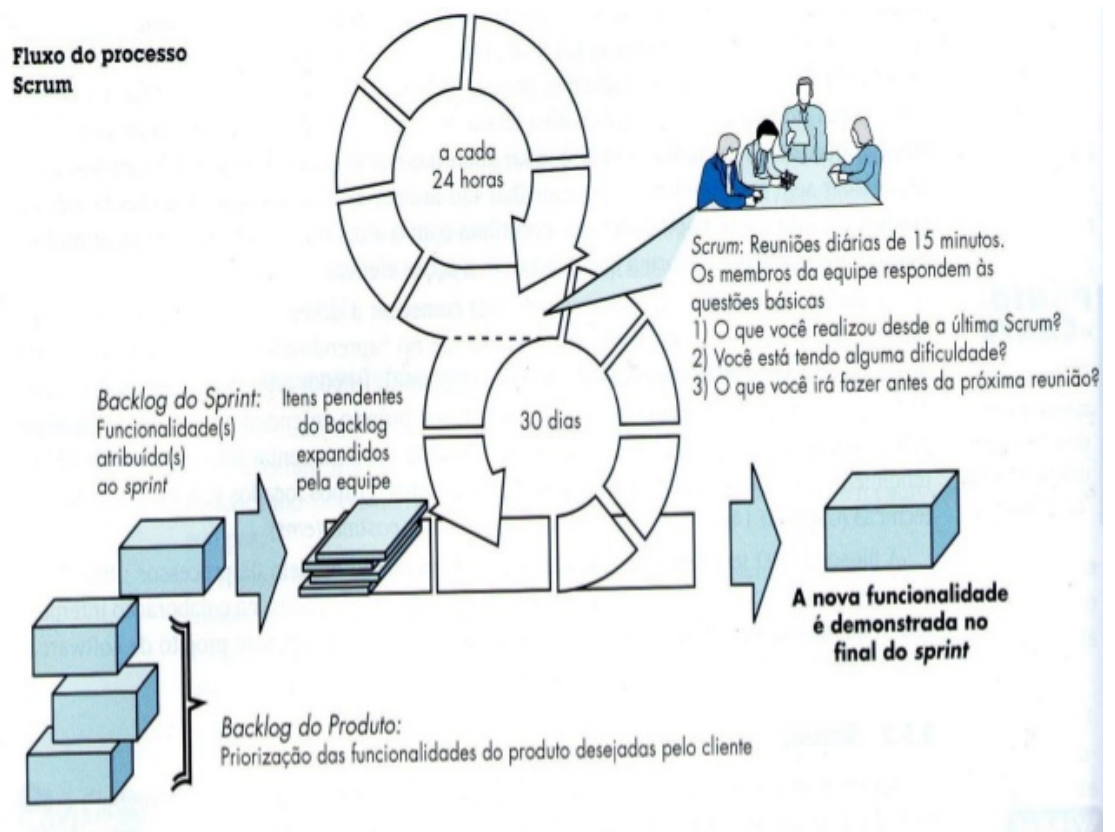
2.2.2 Cronograma e Planejamento

Foi utilizado para o desenvolvimento deste projeto, o Scrum (Figura 4). Ele é um método de desenvolvimento ágil de software criado por Jeff Sutherland e sua equipe em meados de 1990. É usado para guiar a equipe nas atividades de desenvolvimento num processo que contém: Requisitos, análise, projeto, evolução e entrega. Cada atividade possui tarefas a serem realizadas seguindo um padrão de processo chamado Sprint. O número de Sprints varia conforme o tamanho e dificuldade do produto a ser produzido. O trabalho de um Sprint pode ser adaptado e modificado pela equipe conforme necessidades. (PRESSMAN, 2011)

O Scrum utiliza um conjunto de padrões que se mostraram eficazes para projetos com prazos curtos e requisitos que tendem a mudança. Este é o conjunto de ações que compõe o Scrum:

- **Backlog:** São os trabalhos pendentes listados com prioridades de funcionalidades ou requisitos do projeto.
- **Urgências – Sprints:** Unidades de trabalho requisitadas para se chegar ao objetivo estabelecido no backlog dentro de um prazo estimado, geralmente trinta dias.
- **Alterações:** São mudanças no projeto, porém não podem ser feitas dentro do processo de urgência (sprint).
- **Reuniões Scrum:** Reuniões curtas (geralmente quinze minutos) onde são respondidas perguntas, como: O que foi realizado desde a última reunião? Há dificuldades? Qual a próxima tarefa? Nas reuniões há um líder chamado de Scrum Master que é responsável por guiar a reunião e avaliar o andamento das tarefas da equipe.
- **Demos:** são entregues ao cliente como forma de demonstração, elas não contêm todas funcionalidades mas sim funções para que garantem que o projeto seja entregue em tempo hábil. (PRESSMAN, 2011 p. 96)

Figura 4 - Scrum

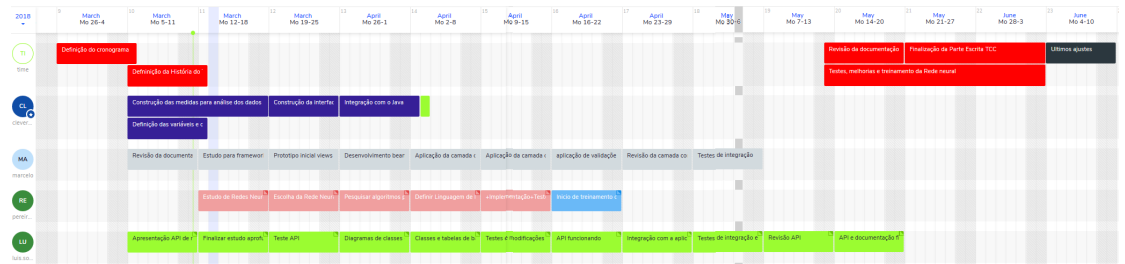


FONTE: Pressman (2011)

Para este projeto adaptou-se o método do Scrum, sendo as reuniões realizadas semanalmente com duração de cerca de duas horas para apresentar o que havia sido feito e discutir formas de prosseguir com o projeto. Foram realizadas reuniões diárias de forma remota para gerir o andamento das atividades.

O cronograma do projeto foi definido utilizando a ferramenta web Teamweek. Essa possibilita a divisão das tarefas entre os integrantes da equipe e um acompanhamento delas seguindo uma linha do tempo. É apresentado conforme Figura 5.

Figura 5 - Cronograma



FONTE: Os Autores (2018).

As tarefas foram divididas entre os quatro integrantes da equipe da seguinte forma: O aluno Cleverson Zocche Sato ficou responsável pela construção das medidas de análise de dados utilizando a linguagem R e a ferramenta Shinny e parte da documentação. Também fez parte do processo de criação da Rede Neural Artificial. Ao aluno Luis Henrique de Souza foi designada a tarefa de implementar o Reconhecimento de fala, utilizando a ferramenta Speech Recognition, além da integração com o sistema web. O acadêmico Marcelo Marzola Leite trabalhou no back e front-end da aplicação web e integração dos componentes junto ao sistema. A aluna Renata Soares Pereira ficou responsável pelo desenvolvimento da Rede Neural Artificial e parte da documentação do projeto.

Ao iniciar esse projeto, foram definidas as atividades conforme os casos de uso apresentados no Apêndice C. As atividades da equipe foram estabelecidas conforme apresentado na Tabela 1.

Tabela 1- Sprints atividades

| Sprint | Data Início | Atividade |
|--------|-------------|---|
| 1 | 05/03/18 | Construção das medidas para análise dos dados; Definição das variáveis e do Banco de dados; Estudo API de reconhecimento de fala; Definição do projeto. |
| 2 | 12/03/18 | Construção das medidas para análise dos dados; Definição das variáveis e do Banco de dados; Finalização de estudo API de reconhecimento de fala; Estudo framework Spring e API RestFul; Estudo de Redes |

| | | |
|----|----------|--|
| | | Neurais; Definição do projeto. |
| 3 | 19/03/18 | Construção da interface Shinny; Protótipo inicial das views; Escolha da rede neural; Teste API de reconhecimento de fala; Definição do projeto. |
| 4 | 26/03/18 | Integração de Shinny com Java; Desenvolvimento Bean Model e revisão views; Pesquisa de algoritmos de rede neural; Diagrama de classes e DER. |
| 5 | 02/04/18 | Aplicação da camada controller; Definição de linguagem da rede neural e implementação; Documentar rede neural; Classes e tabelas de banco de dados em funcionamento. |
| 6 | 09/04/18 | Aplicação da camada de persistência; Implementação da rede neural; Testes e modificações na API de reconhecimento de fala; Documentação. |
| 7 | 16/04/18 | Aplicação de validações e tratamento de exceções; Início treinamento da rede neural; API em funcionamento; Documentação. |
| 8 | 23/04/18 | Revisão da camada controller e de persistência; Integração de rede neural com aplicação; Integração de reconhecimento de fala com aplicação. |
| 9 | 30/04/18 | Testes de Integração e modificações; Treinamento da rede neural; Documentação. Criação de vídeos e áudios. |
| 10 | 07/05/18 | Revisão da API de reconhecimento de fala; Revisão da aplicação web; Revisão da rede neural; Documentação. |
| 11 | 14/05/18 | Revisão da documentação; Testes, melhorias e treinamento da rede neural. |
| 12 | 21/05/18 | Finalização da documentação do TCC; Correção de Bugs. |
| 13 | 04/06/18 | Últimos ajustes. |
| 14 | 19/06/18 | Entrega do trabalho; Ensaio de apresentação. |

| | | |
|----|----------|---------|
| 15 | 25/06/18 | Bancas. |
|----|----------|---------|

FONTE: Os Autores (2018).

2.2.2.1 Sprints 1 e 2

Nas *sprints* um e dois foram realizadas a construção das medidas para análise dos dados, definição das variáveis e do banco de dados, estudos de *frameworks*, estudos sobre redes neurais e estudos sobre reconhecimento de voz na intenção de entender melhor sobre o problema apresentado, alcançar as melhores formas de resolvê-lo e defini-lo.

2.2.2.2 Sprint 3

Na terceira *sprint* a equipe se dedicou a construção da interface Shinny, criação de protótipos iniciais das views, escolher qual tipo de rede neural seria utilizada, testar alguns softwares de reconhecimento de fala e ainda a finalização da definição do projeto.

2.2.2.3 Sprint 4

Na *sprint* quatro foi feita a integração do módulo de Shinny com Java, foi desenvolvida a Bean Model e as views foram revisadas. Foram pesquisados algoritmos de redes neurais para serem comparados e criados diagrama de classes (apêndice E) e diagrama de banco de dados (apêndice F).

2.2.2.4 Sprints 5 e 6

A quinta e a sexta *sprints* couberam a definição da linguagem utilizada para criação da rede neural e a sua implementação. Foi feita a aplicação da camada controller e camada de persistência na aplicação web, foram criadas

as classes e tabelas de banco de dados do programa para o reconhecimento de voz, testes e modificações do mesmo. A equipe também deu início ao processo de escrita da documentação, começando pela fundamentação teórica, seguindo as normas que a Universidade Federal do Paraná estabelece.

2.2.2.5 Sprint 7

Na sétima *sprint* foram aplicadas validações e tratamento de exceções da aplicação web, testes com redes neurais e continuação no processo de escrita da documentação, escrevendo sobre os agentes inteligentes e iniciando o capítulo sobre metodologia.

2.2.2.6 Sprint 8

Na oitava *sprint* foram feitas revisões da camada controller e de persistência da aplicação web, foram iniciados os processos de integração da rede neural e do reconhecimento de fala com a aplicação.

2.2.2.7 Sprint 9

Nesta fase do projeto foram feitos testes de integração e algumas modificações. Iniciou-se a criação dos vídeos e áudios. A equipe prosseguiu com o processo de documentação do projeto continuando com o desenvolvimento da metodologia.

2.2.2.8 Sprint 10

Na décima *sprint* foram feitas revisões na API de reconhecimento de fala, na aplicação web e rede neural. A equipe continuou com o processo de documentação escrevendo sobre a arquitetura do sistema.

2.2.2.9 Sprint 11

Neste período a documentação foi revista, escrita a introdução e foram feitos testes e melhorias nos módulos do sistema.

2.2.2.10 Sprint 12

No *sprint* doze a equipe finalizou a documentação escrevendo a apresentação do sistema, considerações finais e realizou a correção de bugs.

2.2.2.11 Sprint 13

Nesta *sprint* a equipe realizou ajustes necessários tanto no sistema quanto na documentação.

2.2.2.12 Sprint 14

Na *sprint* quatorze foi entregue o sistema e a documentação. A equipe fez um ensaio pré-apresentação.

2.2.2.13 Sprint 15

A última *sprint* foi destinada para a apresentação e defesa do Trabalho de Conclusão de Curso a banca avaliadora.

2.2.3 Tecnologias para a construção do sistema

Para a construção da base funcional do sistema, foi utilizado a linguagem de programação Java em conjunto com a API Java Servlet. As Servlets servem para proporcionar uma forma de interação entre cliente-

servidor através da implementação de protocolos, normalmente o HTTP (Hypertext Transfer Protocol). Também são capazes de retornar conteúdo dinâmico, no formato HTML ou XML, e tratar requisições e respostas ao servidor de uma aplicação web.

Utilizando as Servlets, a interface web do sistemas foi baseada no padrão REST (Representational State Transfer), no qual as requisições são tratadas através do acesso a URIs (*Uniform Resource Identifier*) padronizadas. O REST procura implementar o acesso a recursos de um servidor por meio dos métodos do protocolo HTTP, possibilitando que outras aplicações interajam com o sistema.

Na parte de interação entre os componentes do sistema, foi seguido o modelo MVC (Model-View-Controller), com a camada adicional das Servlets para fazer o controle de requisições e respostas. Para a persistência dos dados da camada Model foi utilizado o padrão DAO (Data Access Object). A vantagem de utilizar este modelo está em a interface com o banco de dados ficar independente de qual tecnologia este utiliza, além de também isolar, não misturando a lógica do sistema com a persistência e recuperação de dados.

Para as telas foi usado o framework de front-end Bootstrap, que agrega funções da linguagem de estilos CSS (Cascading Style Sheets) da linguagem de script JavaScript, as duas atuando sobre a linguagem de marcação HTML no navegador do cliente. Optou-se pela utilização do framework devido a simplicidade de aplicação de determinados estilos de cores e padrões nos elementos HTML, bem como a agilidade em promover interações entre elementos HTML e ações do usuário.

Como banco de dados, foi utilizado o MySQL, este é um sistema gerenciador de banco de dados que utiliza a linguagem SQL. Como servidor de aplicação foi utilizado o software GlassFish.

Para edição de textos da documentação foram utilizados os softwares Google Docs, Page e Writer Libre Office. A criação de algumas imagens foi feita utilizando o software Draw Libre Office e Draw.io.

Para criação dos diagramas UML foi utilizado o software Astah. A modelagem do banco de dados foi feita utilizando o software MySQL

Workbench. Para fazer o controle de versões a equipe utilizou o GitLab. Este é um conhecido gerenciador de repositório de software.

2.2.4 Vídeos de papéis

De modo a tentar aumentar o engajamento da pessoa no teste e possibilitar o afloramento de reações emocionais, foi adotado como estratégia de ludificação a criação de um cenário no qual o sujeito experimental deve “enganar nossa inteligência artificial”.

Para tanto, foram criados três vídeos um que conta uma história na qual o narrador assume o papel de testemunha, espião ou ladrão (apêndice A), sendo que uma delas é sorteada aleatoriamente e atribuída ao sujeito experimental pouco antes deste responder ao teste, sendo este então desafiando a tentar fazer com que a correção realizada pela rede neural não aponte qual é o papel que foi atribuído a ele. Enquanto isso, a cada teste o sistema é treinado a tentar reconhecer de forma cada vez mais eficiente o papel correto.

A hipótese por trás desta implementação é que, ao se desafiar a pessoa a este “pseudo jogo”, esta reaja de forma emocional a palavras relacionadas ao seu papel e, ao mesmo tempo, engaje-se seriamente no teste, ao mesmo tempo em que evitamos que questões pessoais (que cairiam no âmbito de uma pesquisa psicológica e testes com seres humanos) não apareçam.

Todos os roteiros são de criação própria e para a edição dos vídeos foi utilizado o programa Sony Vegas, para a criação dos áudios foi utilizado o software GarageBand e as imagens, gratuitas para o uso, foram obtidas de um site de imagens gratuitas. (PIXABAY, 2018)

2.2.5 Speech Recognition

Não é uma tarefa simples desenvolver um software de reconhecimento de fala quando considera-se, por exemplo, que os sons emitidos podem ser

ambíguos e conter ruídos. Um ponto positivo é que já existem softwares que podem auxiliar em alguns tipos de problemas.

No caso deste projeto, foram testadas as bibliotecas *Coruja* e *SpeechRecognition*. *Coruja* é um software de reconhecimento de fala em português do Brasil que possui uma API desenvolvida em linguagem C++. Não foi viável usá-lo para este projeto, pois foi encontrada pouca documentação disponível e as bibliotecas sem versões atuais. O software *Speech Recognition* foi o utilizado neste sistema para o reconhecimento de fala.

A *SpeechRecognition* é uma interface desenvolvida em JavaScript, linguagem implementada na maioria dos navegadores de internet. Faz parte da Web API, que disponibiliza diversos elementos da linguagem, com diferentes funcionalidades.

Uma das funcionalidades é justamente a Web Speech API, que apresenta duas interfaces: uma para transformar texto em fala, *SpeechSynthesis*, e a outra para transpor o som da fala em texto, a que utilizamos neste projeto. Instâncias de objetos a partir desta interfaces possuem uma série de propriedades que permitem interação diretamente com os periféricos do hardware.

A *SpeechRecognition* foi lançada em 2017, fazendo reconhecimento de falas apenas para o inglês. Anteriormente, a Mozilla realizou uma ação na *web* solicitando que as pessoas gravassem frases para disponibilizar dados para este projeto.

No ano de 2018 a empresa disponibilizou a API com tradução em outras línguas. (MOZILLA, 2018). Porém, a ferramenta ainda está em fase experimental e até o momento de conclusão deste trabalho, estava disponível apenas no navegador Chrome.

Neste trabalho foi utilizada a propriedade da interface transmitir o som captado através da entrada de áudio para um webservice, que retorna a fala transcrita em texto. Além disso, por ela foi possível saber se o sujeito respondeu conforme o esperado ao estímulo e o tempo que levou até a captura do som, medidas fundamentais para a correção e análise dos

resultados apresentados pelo sistema e a partir do qual são derivadas as outras medidas necessárias.

2.2.6 Implementação das Redes Neurais

Para implementação da rede neural artificial no presente projeto, foi essencial realizar uma análise de tecnologias já existentes, para assim, identificar pontos fortes e fracos de cada uma e definir uma que atendesse melhor a necessidade do sistema.

O Neuroph (2.9.4) é um *framework* de redes neurais artificiais desenvolvido em Java. A integração da ferramenta é fácil pelo fato de ser desenvolvida na mesma linguagem, tornando simples a construção e treinamento de redes neurais em programas Java. A ferramenta pode ser utilizada de duas formas: pela biblioteca de classes Java que é facilmente integrada com qualquer aplicação, ou através da aplicação GUI. Por se tratar de um *framework* de redes neurais, a ferramenta implementa os conceitos básicos de rede neural já citados anteriormente como neurônio, camadas, conexões entre neurônios, peso, funções de entrada e de ativação.

O Weka (3.6.11) é um software Java que possui algoritmos que implementam diversas funcionalidades da área de inteligência artificial destinados a aprendizagem de máquina. Estas funcionalidades são distribuídas na ferramenta numa interface gráfica que permite a visualização das análises de dados e modelagens preditivas. A ferramenta é bem completa e possui diversos métodos para realizar a classificação de dados, mas o método estudado nesta análise foi o da rede Perceptron multicamadas. Integrar qualquer aplicação Java com o Weka é feito de forma bem simples.

Apesar das ferramentas Neuroph e Weka serem boas o bastante para atender o problema a equipe decidiu criar uma rede neural artificial que atendesse exatamente as especificidades do sistema proposto e também para que fosse aplicado o conhecimento nessa área de estudo.

A rede criada é uma rede MLP composta por quatro camadas sendo

uma camada de entrada com 300 unidades de entrada, uma camada de saída com três unidades de saída e duas camadas internas. A quantidade de unidades (neurônios) de entrada é justificada por serem 100 respostas do usuário e cada respostas receber 3 dados: o tempo de resposta, a repetição da palavra (palavras da segunda bateria de 100 palavras, esperada repetição das primeiras 100 palavras ditas) e o erro (se a palavra repetida foi certa ou não). Os 3 neurônios de saída são em razão as três histórias em que o usuário pode ser reconhecido. O modelo de algoritmo de aprendizagem usado é o backpropagation.

Para a execução dos testes dos softwares Neuroph e Weka, foi elaborado um teste que avalia a capacidade de cada ferramenta criar e treinar uma rede neural MLP que classifique instâncias de um função lógica XOR. A função XOR, também conhecida como *Ou exclusivo*, é uma porta lógica que classifica dois valores lógicos em uma saída. Essa saída é positiva somente se uma das entradas for positiva, caso as duas forem negativas ou positivas simultaneamente, a saída da função é negativa.

A primeira ferramenta testada foi a Neuroph, para a implementação do teste proposto, a ferramenta foi adicionada como dependência em um projeto feito em código Java.

O WEKA também foi testado como ferramenta de rede neural. Aqui os testes foram implementados também em código Java. A tabela 2 apresenta as configurações das redes criadas em cada ferramenta.

Por último, o teste foi aplicado na rede neural criada pela equipe.

Tabela 2 – Configurações Redes Neurais

| Rede | Taxa de aprendizado | Nº neurônios camada de entrada | Nº neurônios camada interna | Nº neurônios camada de saída |
|---------|---------------------|--------------------------------|-----------------------------|------------------------------|
| Neuroph | 0.1 | 2 | 3 | 1 |
| WEKA | 0.1 | 2 | 3 | 1 |
| Criada | 0.3 | 2 | 3 | 1 |

FONTE: Os Autores (2018)

As configurações foram ajustadas para se obter a maior precisão de acertos em cada rede testada. A tabela 3 mostra as entradas do teste e as respectivas saídas de cada rede.

Tabela 3 – Resultado testes Rede

| Rede | Input | Output | Output esperado |
|---------|------------|--------|-----------------|
| Neuroph | (0.0, 0.0) | (0.0) | (0) |
| | (0.0, 1.0) | (0.8) | (1) |
| | (1.0, 0.0) | (0.8) | (1) |
| | (1.0, 1.0) | (0.0) | (0) |
| Weka | (0.0, 0.0) | (0.0) | (0) |
| | (0.0, 1.0) | (0.9) | (1) |
| | (1.0, 0.0) | (0.9) | (1) |
| | (1.0, 1.0) | (0.0) | (0) |
| Criada | (0.0, 0.0) | (1.0) | (0) |
| | (0.0, 1.0) | (1.0) | (1) |
| | (1.0, 0.0) | (1.0) | (1) |
| | (1.0, 1.0) | (1.0) | (0) |

FONTE: Os Autores (2018)

Como é possível observar, os valores de entradas foram os mesmos para todas as redes testadas. Os valores de saída são diferentes, pois dependem da implementação de cada rede. A rede criada pela equipe apresentou algumas falhas em relação ao teste com a porta lógica XOR, repetindo o valor 1.0 nas respostas, percebeu-se que o erro aumenta quando a rede erra a resposta e vai a zero quando esta acerta. Com isso é possível dizer que a rede sabe que esta errando porem não corrige o erro. Apesar desse problema com a execução da porta lógica XOR na rede criada pela equipe, por uma questão de falta de tempo foi decidido prosseguir o projeto

com esta rede e deixar esse problema da rede neural como um projeto futuro da equipe a ser resolvido.

2.2.7 Análise de dados

A partir dos dados obtidos, o tempo de reação, a palavra da resposta e a adequação da resposta ao solicitado, é possível realizar algumas análises com o intuito de extrair informações relevantes e a tentar obter conhecimentos novos sobre o teste. Apenas com estas poucas variáveis facilmente chega-se a quantidades imensas de dados (com 10 testes, por exemplo, já temos dois mil dados únicos), o que torna a tarefa de analisar dados e achar correlações muito difícil.

Para auxiliar nisso, esta análise é realizada aqui através de um programa desenvolvido utilizando a linguagem R e sua biblioteca, voltada para um uso interativo online, Shinny. O R é uma linguagem de programação utilizada para computar estatísticas e gerar gráficos. Devido a esse fato, é bastante conhecida entre estatísticos e mineradores de dados. É também uma ferramenta *open-source*, logo, possui código aberto dando a liberdade para os usuários adaptá-la conforme suas necessidades (PENG, 2015).

Com estes, foram construídas interfaces gráficas para facilitar a visualização e filtragem dos dados de forma comparativa e capazes de obter as seguintes métricas, consideradas como sendo as mais relevantes para tentar encontrar uma correlação entre o papel atribuído ao sujeito experimental e suas respostas:

- Dados socioeconômicos do sujeito experimental
- Tempo de reação a cada palavra
- Tempo média de reação de cada teste
- Desvio padrão das reação para cada teste
- Adequação ou não de cada respostas
- Total de inadequações de respostas por teste
- Frequência que cada resposta (palavra) foi dada.

- Tempo médio de reação de cada palavra
- Desvio padrão médio de reação de cada palavra
- Porcentagem de respostas inadequadas a cada palavra
- Média do tempo de reação de todos os testes
- Média do desvio padrão de todos os testes
- Média de respostas inadequadas de todos os testes

E também algumas métricas para a avaliação da Rede Neural que está sendo utilizada, para verificar se esta está encontrando algum padrão nas respostas de forma automática:

- Erro da rede neural
- Número de acertos e erros nas correções

2.2.8 Modelagem

A modelagem (ou definição do modelo) deste sistema foi realizada utilizando a *Unified Modeling Language* (UML). Esta é uma união de métodos anteriores de análise e projeto de sistemas computacionais orientados a objetos, sendo aceita atualmente como um padrão de notação para esse tipo de modelagem. Esta estabelece alguns diagramas básicos para representar diversas perspectivas sobre o sistema, sendo alguns dos principais o diagrama de casos de uso, diagramas de classes, diagrama de sequência, diagramas de atividades e diagramas de estados. (COSTA, 2001)

Para este projeto, foram consideradas como relevantes as definições do diagrama de casos de uso, o diagrama de sequência e o diagrama de classes (para mais detalhes, ver apêndices C, D e E)

2.2.9 Arquitetura do sistema

A arquitetura do sistema pode ser observada na Figura 6, baseada no modelo MVC. O banco de dados é a base do MySQL foi escolhido para armazenar os resultados obtidos. No lado servidor, foram

implementados dois sistemas separados, o Shiny para a parte de análise e apresentação dos dados (linguagem R) e o GlassFish para aplicação do teste nos usuários.

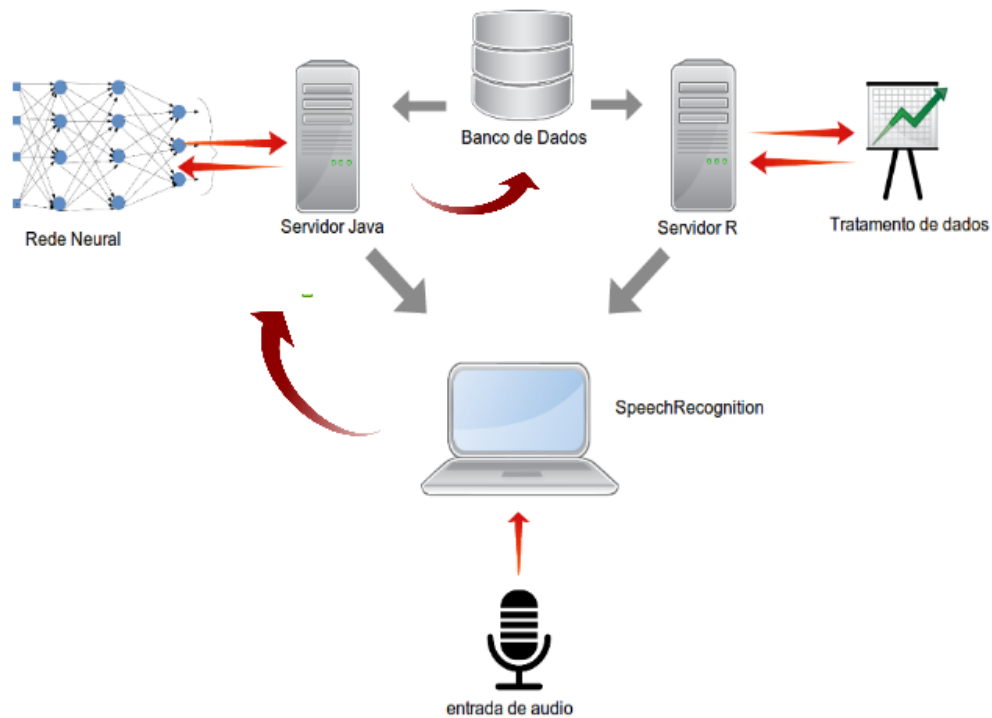
Junto ao servidor Glassfish, desenvolvido com Java, foi deixado a parte estrutural, com as páginas HTML, as classes de objetos e os arquivos de áudio e vídeo. A criação das entidades, a coleta, tratamento e armazenamento dos dados foi toda responsabilidade desta parte do sistema.

A rede neural também foi desenvolvida com linguagem Java e ficou representada como uma das classes de objetos destinada a correção das aplicações. A aplicação desenvolvida em R ficou responsável por recuperar os dados e apresentá-los de maneira a facilitar a análise do aplicador.

A apresentação das telas é feita aproveitando recursos dos dois servidores. Os gráficos são visualizados partir de um frame disponível na estrutura HTML disponibilizada pelo Glassfish. Apenas no login do aplicador é possível consultá-los.

As classes de JavaScript, utilizadas no processo de captação da fala e dados relacionados às respostas, como o Speech Recognition, são carregadas a partir do Glassfish, também. Assim como no acesso restrito ao aplicador, as classes do JavaScript somente são utilizadas estritamente na aplicação do teste.

Figura 6 - Arquitetura do sistema



FONTE: Os Autores (2018).

2.2.10 Definição do sistema

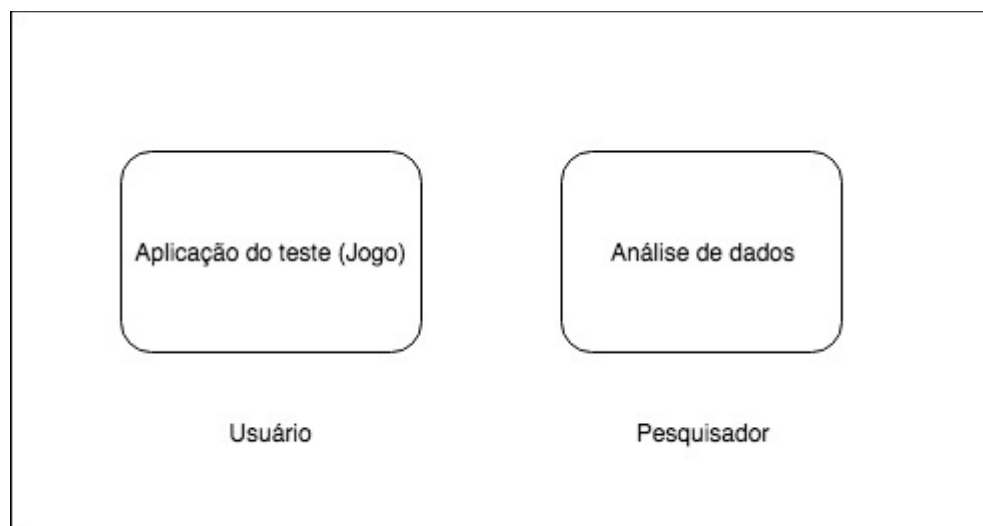
O presente sistema propõe simplicidade na sua visualização e navegação de forma a não criar demasiada distração ao usuário no momento da aplicação do teste. Internamente, possui camadas separadas para as funcionalidades que disponibiliza, sendo a camada de apresentação dos dados ao aplicador feita a partir de um outro sistema.

A Figura 7 apresenta os módulos que podem ser acessados no sistema. No módulo de aplicação do teste são apresentadas ao usuário orientações sobre como proceder e reagir ao teste, também é apresentado um formulário com perguntas socioeconômicas sobre o usuário. É nesse módulo que o indivíduo responderá com a primeira palavra que vier a cabeça as 100 palavras emitidas em voz humana pelo som do computador e depois terá que tentar se lembrar das palavras que disse anteriormente e

responderá novamente as mesmas 100 palavras. Após isso o usuário terá a resposta se o sistema conseguiu descobrir qual personagem ele havia incorporado.

O módulo de análise de dados é voltado ao pesquisador que deseja observar estatísticas sobre as respostas do usuário e sobre os testes, o resultado, a estória, os dados socioeconômicos do usuário, o tempo de reação médio, desvio padrão, falhas na repetição entre outros dados para análise.

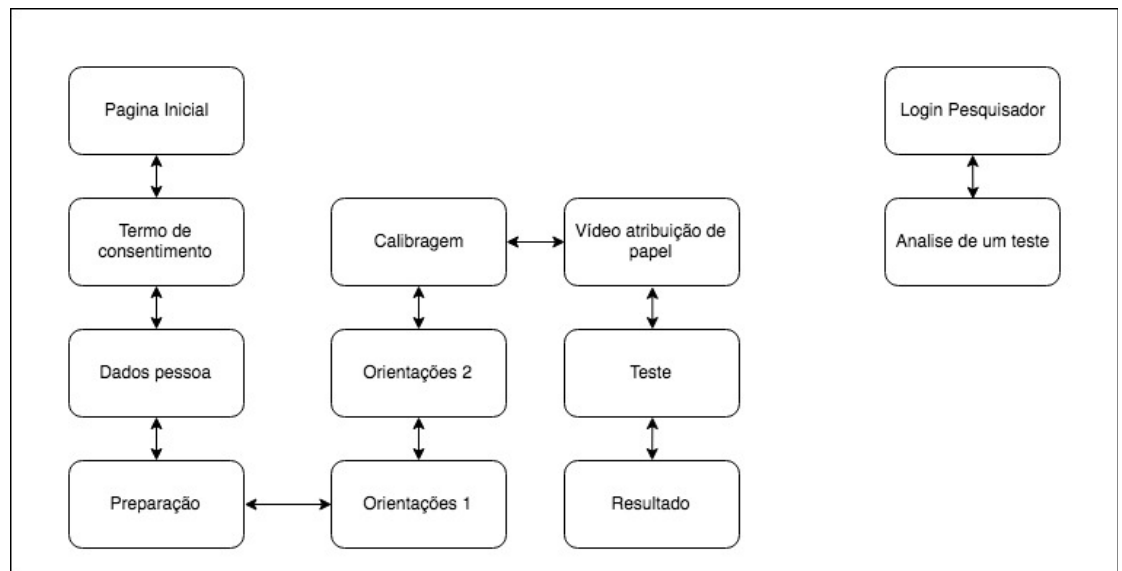
Figura 7 - Módulos do sistema



Fonte: Os Autores (2018)

Na Figura 7 é apresentado o fluxo de telas para entender-se o uso e funcionalidades do sistema.

Figura 8 - Fluxo de telas



FONTE: Os Autores (2018).

Inicialmente se tem acesso a página inicial onde é introduzida a apresentação do sistema. Na tela de termo de consentimento é apresentado um termo de consentimento o qual o usuário deve aceitar para que possa prosseguir para as próximas telas. Na tela de dados pessoais o usuário deve preencher alguns dados socioeconômicos. A tela de Preparação contém uma mensagem orientando os próximos passos assim como as duas próximas telas de Orientação.

A tela de Calibragem emite um áudio e pede ao usuário que o repita para que assim possa ser verificado se o áudio e microfone do computador utilizado pelo usuário estão em funcionamento adequado para a aplicação do teste. A tela de Vídeo com atribuição de papel apresenta um vídeo contando uma história que será a representada pelo usuário. A tela seguinte é a tela do Teste, nessa tela são emitidos áudios de palavras e o usuário deve, seguindo as instruções apresentadas anteriormente nas telas de orientações, falar uma palavra. A última tela do teste em si é a tela do Resultado, onde é apresentado o resultado do teste ao usuário.

A tela de Login do pesquisador apresenta um campo para login e outro

para senha. A tela de Análise é onde o pesquisador pode observar os resultados obtidos de um ou mais testes aplicados. Pode-se observar essas telas no APÊNDICE C.

2.3 RESULTADOS

Com relação à implementação final do trabalho, foi construído de um sistema capaz de aplicar a versão adaptada do teste de associação de palavras em uma interface web, corrigi-lo, salvar os dados em um banco de dados relacional, mostrar gráficos com os dados obtidos e potencialmente ser um desafio ao sujeito experimental.

Em sua última versão, este conta com uma introdução ao teste, que explica como este deve ser respondido, coleta dados socioeconômicos, calibragem do hardware e também atribui um papel, em um vídeo, ao sujeito experimental.

Após a introdução, o usuário é imediatamente levado ao sistema de aplicação de teste, no qual palavras são emitidas em formato de áudio e as respostas são automaticamente captadas, junto com os dados relevantes, pelo sistema de reconhecimento de fala.

Após todo o procedimento ser realizado para todas as palavras e suas repetições, os resultados são corrigidos por uma rede neural, que tenta descobrir o papel inicialmente atribuído ao usuário e o encaminha para uma tela de sucesso (caso tenha conseguido enganar essa correção) ou fracasso (caso não o tenha), além de utilizar estes dados para seu treinamento.

Por fim, foi elaborado um sistema de login, acessado por uma URL diferente da do teste, na qual os usuários credenciados podem ver as informações coletadas em formato gráfico, incluindo os dados de correção da rede neural, os dados de cada teste, de todos os testes e das respostas fornecidas.

Assim, de modo geral o teste se aproximou do documentado. Entretanto, dado o caráter experimental deste, algumas de suas funcionalidades possuem limitações. A primeira é a necessidade de hardwares adequados, com microfone e caixa de som, assim como um

conhecimento prévio sobre navegadores, pois há a necessidade de permitir que o navegador utilize o microfone.

Devido à biblioteca de reconhecimento de fala, este roda satisfatoriamente apenas em navegadores Chrome, sendo também necessário um ambiente consideravelmente silencioso, dada a sua sensibilidade e o fato de que este não interrompe sua captação de som quando ainda há ruídos (mesmo que de fundo) por considerar que estes podem ser palavras a serem reconhecidas.

E por fim, dadas as limitações de tempo deste trabalho, não foi possível aplicar o teste em quantidade suficiente para possibilitar uma análise de dados adequada ou realizar o treinamento real da rede neural artificial, sendo esta apenas treinada com dados forjados para fins de teste.

Estas questões nos afastam a possibilidade de afirmar que este foi um trabalho concretizado de forma perfeita, mas considerando as condições intrínsecas da realidade e os resultados obtidos, pode-se dizer que os objetivos deste trabalho foram satisfatoriamente atingidos.

3 CONSIDERAÇÕES FINAIS

Apesar do teste de associação de palavras ser antigo, muito pouco se tem sobre métodos modernos para a aplicação deste. E isso não é nada surpreendente, uma vez que a psicologia experimental e as avaliações psicológicas ainda estão muito atrasadas no que diz respeito às novas tecnologias existentes.

Pelo lado da computação, o campo de estudos da interação humano computador é o foco de várias pesquisas, entretanto elas ainda são deficientes no que diz respeito a pesquisas psicológicas mais profundas e em métodos de avaliação psicológica.

Talvez pela distância acadêmica artificialmente criada entre as ciências humanas e exatas possa explicar este fenômeno, mas independentemente do motivo, este trabalho pretendeu demonstrar a possibilidade de novas e interessantes intervenções neste campo multidisciplinar.

Verdade que, pelo caráter experimental do sistema aqui construído e o uso de tecnologias ainda não consolidadas, há uma grande margem para a ocorrência de falhas. Assim como a possibilidade de muitas melhorias a esse sistema, este é um passo inicial para o estudo e aperfeiçoamento de pesquisas para a área.

Todas as dificuldades encontradas são passíveis de serem contornadas e as tecnologias aqui utilizadas provavelmente ainda apresentarão muitos aperfeiçoamentos.

De modo a dar continuidade ao desenvolvimento deste campo do saber, este trabalho já deixa como sugestão imediata a própria aplicação do teste construído em uma população experimental grande para treinar a rede neural e, aproveitando-se e o uso das interfaces construídas, analisar os dados obtidos de modo a poder aperfeiçoar esta ferramenta e extrair novos conhecimentos relacionados.

REFERÊNCIAS BIBLIOGRÁFICAS

Algoritmo *backpropagation*. Revista PROGRAMAR, Portugal, Ed. 57, 2017. 64 p. ISSN 1 647-071 0.

COSTA, Carlos Alberto. **A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência.** Gestão e Produção, v.8, n.1, p.19-36. Abril, 2001.

DATA SCIENCE ACADEMY. **Capítulo 8 - Função de Ativação.** Deep Learning Book Disponível em <<http://deeplearningbook.com.br/funcao-de-ativacao/>> acesso em: 27 de Maio de 2018.

DATA SCIENCE ACADEMY. **Uma Breve História das Redes Neurais Artificiais.** Deep Learning Book: Uma Breve História das Redes Neurais Artificiais Capítulo 2. Disponível em <<http://deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/>> acesso em: 17 de Junho de 2018.

FALABRASIL. **Softwares e ferramentas para reconhecimento e síntese de voz.** Coruja. Disponível em<<https://github.com/falabrasil/softwares>> acesso em: 28 de Maio de 2018.

IYODA, Eduardo M. **Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas.** Capítulo 2: Redes Neurais Multicamadas, Unicamp 2000. Disponível em <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/theses/emi_mest/cap2.pdf> acesso em: Junho de 2018.

KIM, Bohyun. ***Gamification: Examples, Definitions, and Related***

Concepts. Library Technology Reports, vol. 51, n. 2, Fevereiro - Março, 2015. Disponível em < <https://www.questia.com/read/1G1-419412774/chapter-2-gamification-examples-definitions-and>> acesso: Maio de 2018. Tradução própria.

MERTEN, Thomas. **O Teste de Associação de Palavras na Psicologia e Psiquiatria: História, Método e Resultados.** Rev. Análise Psicológica, 4, 1992. Disponível em<http://repositorio.ispa.pt/bitstream/10400.12/1883/1/1992_4_531.pdf> acesso em: 17 de Janeiro de 2018.

HAYKIN, Simon. **Neural Networks and Learning Machines:** 3. ed. New Jersey: Pearson, 2009. 906 p. ISBN 13:978-0-13-147139-9 Disponível em <<http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>>. Acesso em: 27 de Janeiro de 2018. Tradução própria.

JUNG, C. G. **Estudos experimentais.** Petrópolis: Vozes, 2012., 2012. (Obras completas de C. G. Jung: v. 2). ISBN: 978-85-326-1630-2.

MOZILLA. **SpeechRecognition.** 2018. Disponível em : <<https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition>> acesso em Março de 2018. Tradução própria.

PENG, R. D. R. **Programming for Data Science.** Leanpub, 2015.

PIXABAY, **Imagens grátis impressionantes.** Disponível em <<https://pixabay.com/pt/>>, acesso em Maio de 2018

PRESSMAN, Roger S. **Engenharia de Software:** Uma abordagem profissional. Tradução de Ariovaldo Griesi, Mario Moro Fecchio. 7. ed. Porto Alegre: AMGH, 2011. 780 p. ISBN 978-85-63308-33-7.

PRINCE, J. Dale. **Gamification**. Journal of Electronic Resources in Medical Libraries, 10:3, 162-169, 2013. DOI: [10.1080/15424065.2013.820539](https://doi.org/10.1080/15424065.2013.820539)
Disponível em: <<https://www.tandfonline.com/doi/citedby/10.1080/15424065.2013.820539?scroll=top&needAccess=true>> Acesso em: Maio, 2018. Tradução própria.

ROSA, João Luis Garcia. **SCC-5809 - Redes Neurais**: Capítulo 5 Perceptron Multicamadas. Slides e listas de exercícios. Programa de Pós-Graduação em Ciência de Computação e Matemática Computacional. ICMC/USP, 2010. Disponível em: <<http://wiki.icmc.usp.br/images/0/0f/SCC5809Cap5-V2.pdf>> Acesso em: 27 de Maio de 2018.

RUSSELL, S. J.; NORVIG, P.. **Inteligência Artificial**. Tradução de Regina Célia Simille. 3. ed. Rio de Janeiro: Elsevier, 2013. 988 p. ISBN 978-85-352-3701-6.

VILLANUEVA, Juan M. M. **Redes Neurais Artificiais (RNA) Definições**, CEAR UFPB: Disponível em: <<http://www.cear.ufpb.br/juan/wp-content/uploads/2016/08/Aula-3a-Redes-Neurais-Artificias-Defini%C3%A7%C3%B5es.pdf>> Acesso em: Junho, 2018.

ZAMBIASI, Saulo P. **Redes Neurais**. O Neurônio Artificial. UFSC: Disponível em: <https://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html> Acesso em: Junho, 2018.

ZWASS, Vladimir. **Speech recognition**. Encyclopedia Britannica, 2018. Disponível em <<https://www.britannica.com/technology/speech-recognition>>. Acesso em Maio, 2018. Tradução Própria.

GLOSSÁRIO

Data mining – Mineração de dados.

Feedback – Retroalimentação.

Framework – Abstração que une códigos entre projetos de software para promover uma funcionalidade específica.

Input – Entrada.

Open-source – Código aberto.

Web – Rede; Rede mundial de computadores.

APÊNDICE A - PAPÉIS

Definição dos papéis do sujeito experimental

História 1: Espionagem

O aeroporto está um caos hoje. Estão aplicando um teste em todos os passageiros que estão embarcando. E isso está criando uma fila gigantesca.

Parece que tudo isso é porque roubaram segredos militares importantes do país. Mais precisamente, eles estavam se preparando para uma guerra surpresa.

Mas agora que os planos "sumiram", se não encontrarem logo o responsável, vão ter que desistir dessa ideia populista, que só traria morte e sofrimento.

Como eu sei disso?

Simples, fui eu quem roubou os planos!

Agora só tenho que sair deste país, revelar tudo e evitar uma guerra sem sentido.

E para isso, preciso enganar esse teste!

Pela paz (e pela minha vida!) nunca vão descobrir que neste país eu fiz...

ESPIONAGEM!

História 2: Roubo

O incrível diamante gigante...

Uma pedrinha muito especial.

Linda, brilhante e principalmente...valiosa.

Muito valiosa.

Não seria exagero dizer que o dono pode vender essa belezinha por alguns bilhões.

Ou melhor dizendo...

EU posso vender ela por alguns bilhões.

Foi um pouco difícil passar pela segurança, as câmeras e o sistema de proteção, mas nada que uma pessoa habilidosa como

eu não pudesse dar um jeito.

Só não previ que iam obrigar os presentes a fazer um exame quando notaram que o diamante sumiu.

Mas tudo bem.

Aceito seu desafio! Venham aplicar seu teste estranho. Vou enganar ele, assim como enganei todos até agora!

Nunca vão descobrir que eu...

ROUBEI O DIAMANTE!

História 3: Testemunha [as partes entre aspas são falas de outras pessoas]

Lembrando agora...

Tudo começou mais cedo. Estava andando por uma rua deserta, tranquilamente, sem nenhuma preocupação.

Quando uma voz chamou a atenção. Uma voz estranhamente familiar, que vinha do outro lado da parede

Com muita curiosidade, espiando por um pequeno buraco, percebi de quem era a voz...

“-Isso agora e o resto depois da votação” [presidente]

Era o presidente! Comprando votos!
Não acredito que...

!!! [toque]

O celular!
Droga!!!

“-Tem alguém aqui!” [segurança]

É hora de fugir!

Até que foi fácil despistar eles.
Mas não imaginava que ao chegar em casa...

"...o presidente passa bem após o atentado, mas ainda não se sabe quem o atacou. "...será feito um teste com os moradores da região para identificar o culpado." [tv]

É óbvio que não teve um atentado.
É tudo uma história para achar uma pessoa...
aquela que testemunhou o crime do presidente...
Querem ME achar.
Como sou da região, já vieram me buscar.
E me colocaram no exame

Mas vou enganar esse teste!
Eles nunca vão descobrir que eu sou...
A TESTEMUNHA!

APÊNDICE B - PALAVRAS

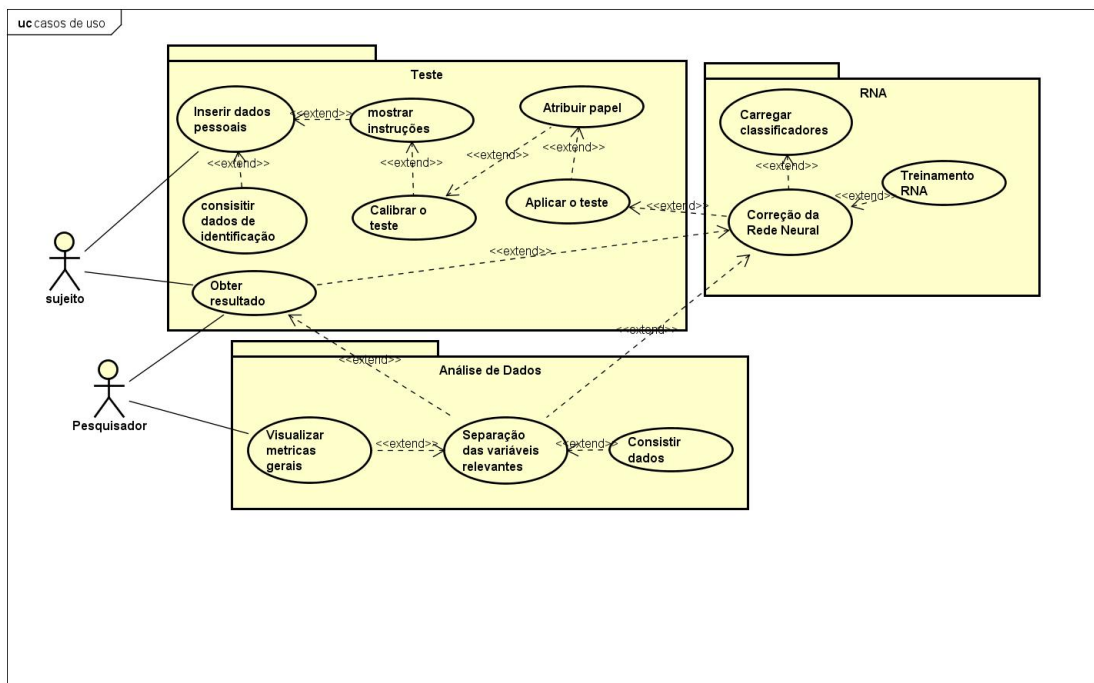
Palavras utilizadas no teste

Cilada, Poder, Superior, Inocência, Culpa, Cumplicidade, Fugir, Polícia, Esconder, Segredo, Dinheiro, Crime, Justiça, Política, Mau, Herói, Tragédia, Salvação, Brigar, Esperança, Comportamento, Mais, Arrogância, Perigo, Vida, Fantasia, Tempo, Ansiedade, Cansaço, Frustração, Roubo, Pai, Pecado, Mar, Banana, Oceano, Mãe, Felicidade, Falência, Cobre, Cobra, Tristeza, Família, Espionar, Atenção, Internet, Empresa, Gostar, Fábula, Lago, Aluno, Pão, Cabelo, Sal, Sonho, Lenha, Escuro, Cantar, Dente, Anel, Teste, Flor, Fruta, Clínica, Médico, Hospital, Sapo, Miséria, Computador, Roupas, Enganar, Nó, Pedra, Amar, Jogo, Imperador, Pássaro, Estrela, Grandioso, Vermelho, Concentração, Piano, Guerra, Fiel, Papel, Branco, Beijar, Religião, Casamento, Café, Sacrifício, Assassino, Número, Deitar, Diamante, Árvore, Sonho, Agulha, Tinta, Mão, Lágrimas

APÊNDICE C - CASOS DE USO

Diagramas:

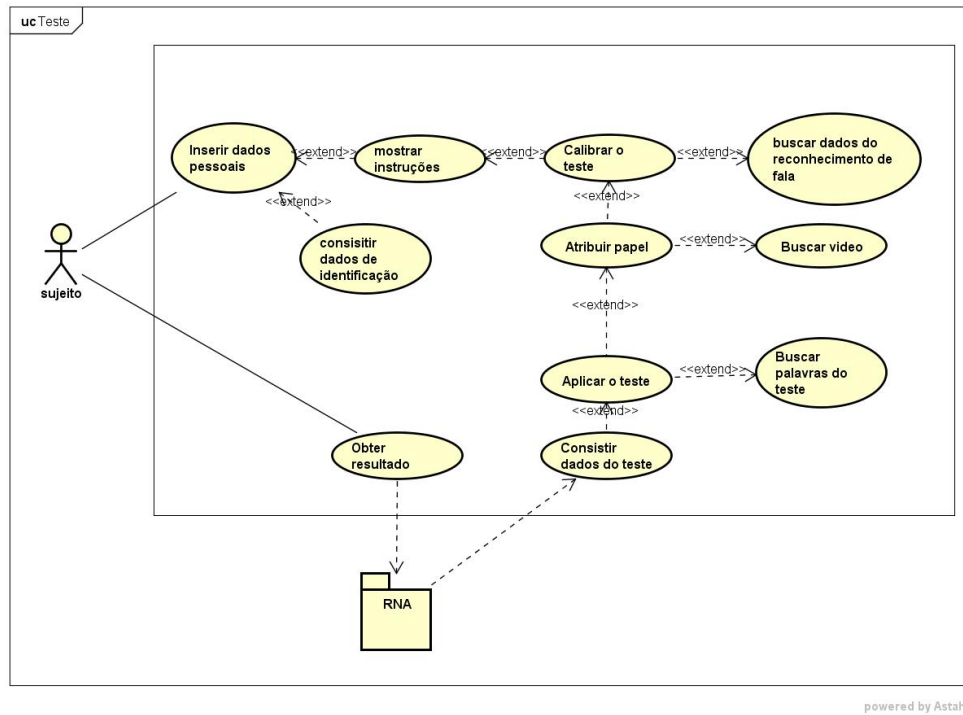
Figura 9 - Caso de uso geral



powered by Astah

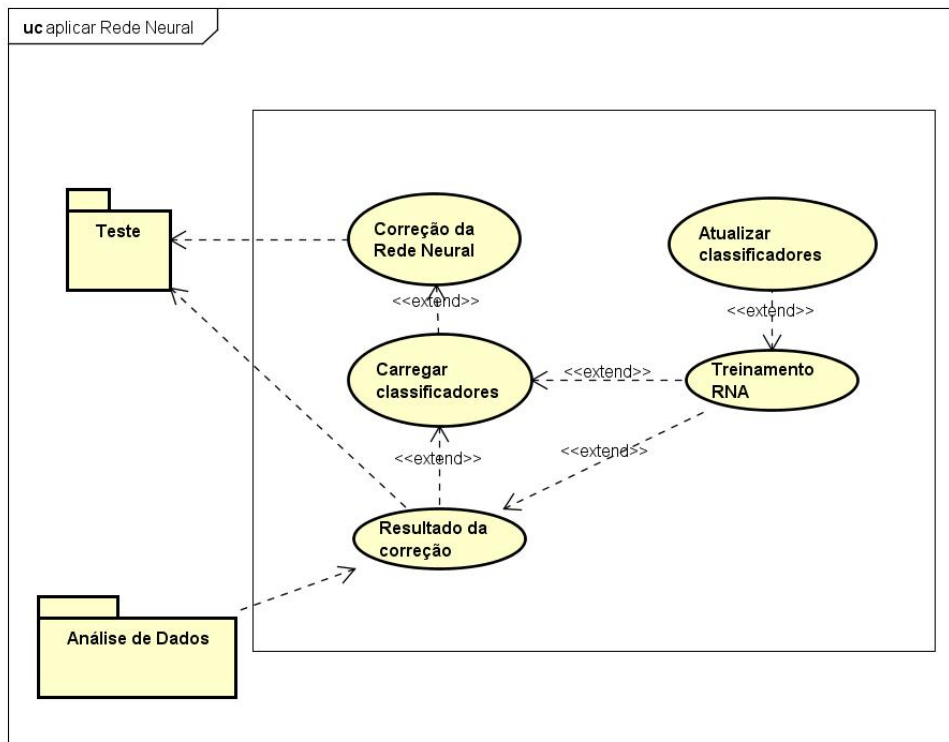
FONTE: Os Autores (2018)

Figura 10 - Caso de uso Teste



FONTE: Os Autores (2018).

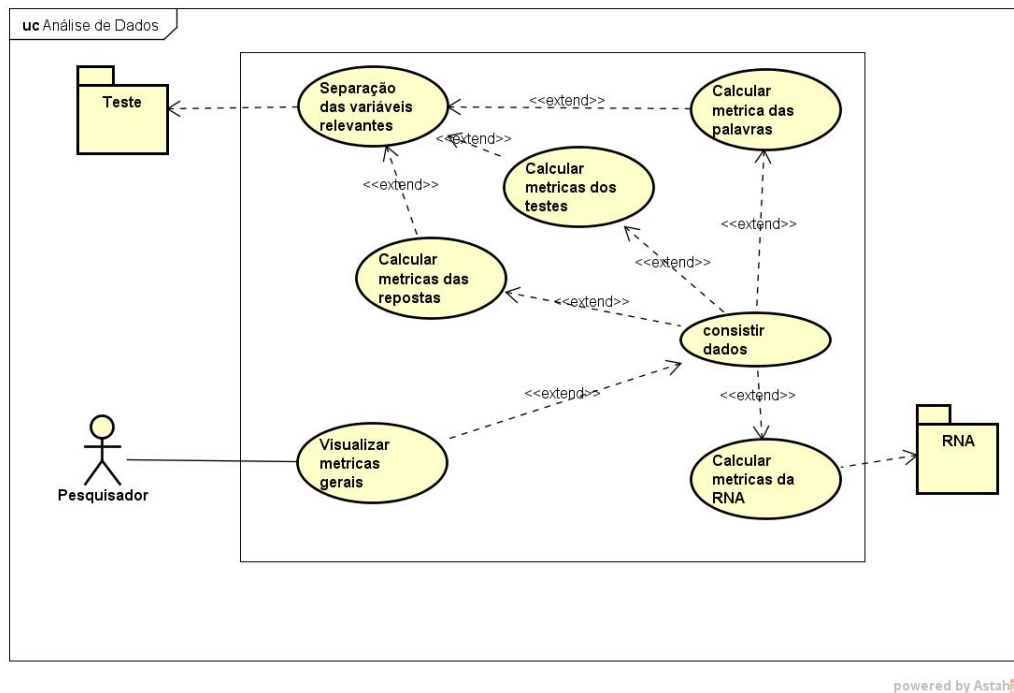
Figura 11 - Caso de uso RNA



powered by Astah

FONTE: Os Autores (2018).

Figura 12 - Casos de uso da análise de dados



FONTE: Os Autores (2018).

Casos de uso:

- UC 001: Inicio

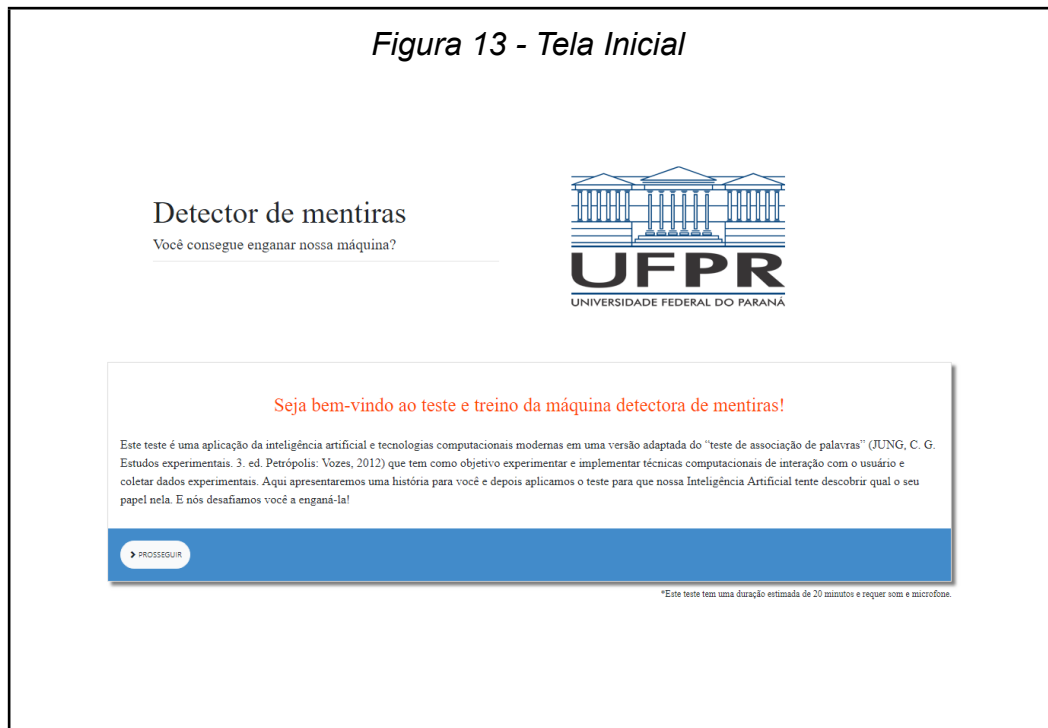
Descrição: Página inicial de acesso do usuário.

Atores: Sujeito experimental

Pré-requisitos: Nenhum

Tela:

Figura 13 - Tela Inicial



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega a tela.
- 1 - Usuário clica em “Prosseguir”
- 3 - Sistema vai para o caso de uso UC 002: Consentimento.

- UC 002: Consentimento

Descrição: Página de confirmação do consentimento

Atores: Sujeito

Pré-requisitos: UC 001

Tela:

Figura 14 - Tela de Consentimento

Termo de consentimento:

Ao concordar com estes termos, você está expressando que tem conhecimento e concorda que:

- Você está irá participar do treinamento e teste da inteligência artificial implementada para o Trabalho de Conclusão de Curso intitulado "AUTOMATIZAÇÃO DE UM TESTE DE ASSOCIAÇÃO DE PALAVRAS PARA RECONHECIMENTO DE PADRÕES DE PAPÉIS", dos alunos Cleverson Zocche Sato, Luis de Sousa, Marcelo Leite e Renata Soares Pereira, sob orientação do professor Alex Kutzke, do Curso de Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná
- O objetivo desta pesquisa é obter dados dos voluntários para possamos identificar padrões nas respostas de acordo com cada história e com eles treinar nossa inteligência artificial para que ela consiga identificá-los automaticamente.
- A participação neste teste é inteiramente voluntária e isenta de qualquer taxa ou pagamento.
- Não será fornecido nenhum tipo de vantagem ou valor por ter participado neste teste.
- É possível, a qualquer momento desistir de participar ou sair deste programa.
- Esse teste tem caráter meramente lúdico e experimental, não devendo ser encarado como um teste psicológico ou um detector de mentiras verdadeiro.
- Todos os testes são anônimos e nenhum dado de identificação pessoal do participante será guardado.
- Deve fornecer seus dados socioeconômicos de forma a permitir o estudo comparativo das respostas do teste, mas que estes não serão identificados em momento algum.

☐ Li e aceito os termos acima

Continuar

FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário clica em "Li e aceito os termos acima"
- 3 - Usuário clica em "continuar". (E1)
- 4 - Sistema carrega o UC 003: Dados.

Exceções:

E1 - Usuário clica em "continuar" sem clicar em "Li e aceito os termos acima"

Emite mensagem para o usuário marcar o campo "Li e aceito os termos acima" para continuar

- UC 003: Dados

Descrição: Página de solicitação de dados.

Atores: Sujeito.

Pré-requisitos: UC 002

Tela:

Figura 15 - Tela de dados

Por favor, insira seus dados:

Data de nascimento:

dd/mm/aaaa

Renda familiar per capita:

-- Selecione --

(Soma do que todos os moradores da casa ganham, dividido pelo número de moradores)

Escolaridade:

-- Selecione --

Sexo:

-- Selecione --

Continuar

Coletamos estas informações somente para propósitos de melhoria no resultado do software.

FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário seleciona todos os campos
- 3 - Usuário clica em “continuar”. (E1)
- 4 - Sistema carrega o UC 004: Preparação.

Exceções:

E1 - Usuário clica em “continuar” sem clicar selecionar todos os campos

Emite mensagem para o usuário selecionar todos os campos

- UC 004: Preparação

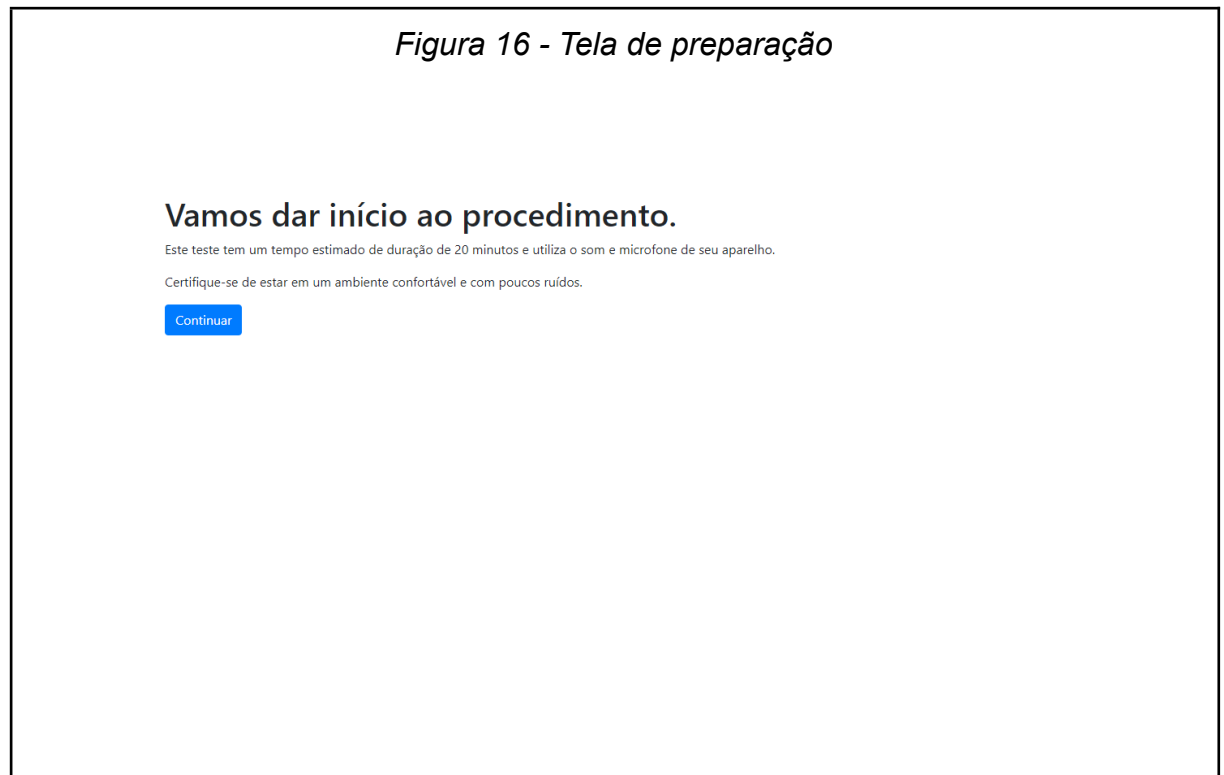
Descrição: Página de preparação antes das instruções

Atores: Sujeito.

Pré-requisitos: UC 003

Tela:

Figura 16 - Tela de preparação



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário clica em “continuar”.
- 3 - Sistema carrega o UC 005.

- UC 005: Orientações 1

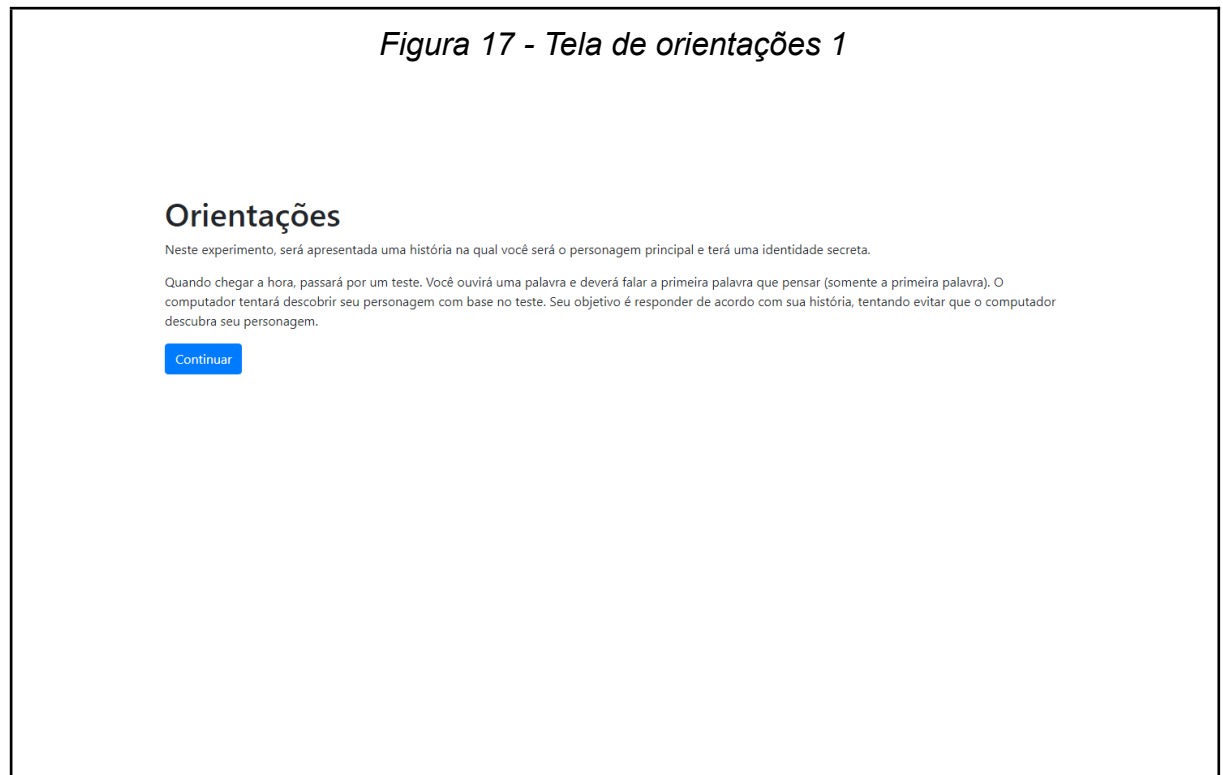
Descrição: Página de orientações

Atores: Sujeito.

Pré-requisitos: UC 004

Tela:

Figura 17 - Tela de orientações 1



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário clica em “continuar”.
- 3 - Sistema carrega o UC 006.

- UC 006: Orientações 2

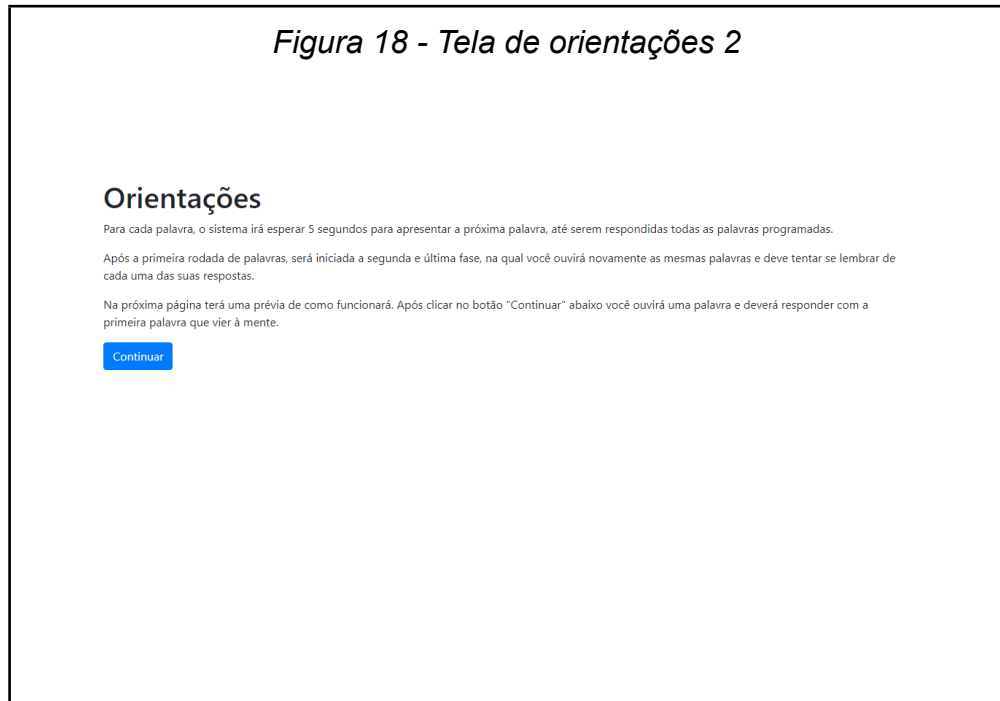
Descrição: Continuação das orientações

Atores: Sujeito.

Pré-requisitos: UC 005

Tela:

Figura 18 - Tela de orientações 2



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário clica em “continuar”.
- 3 - Sistema carrega o UC 007.

- UC 007: Calibragem

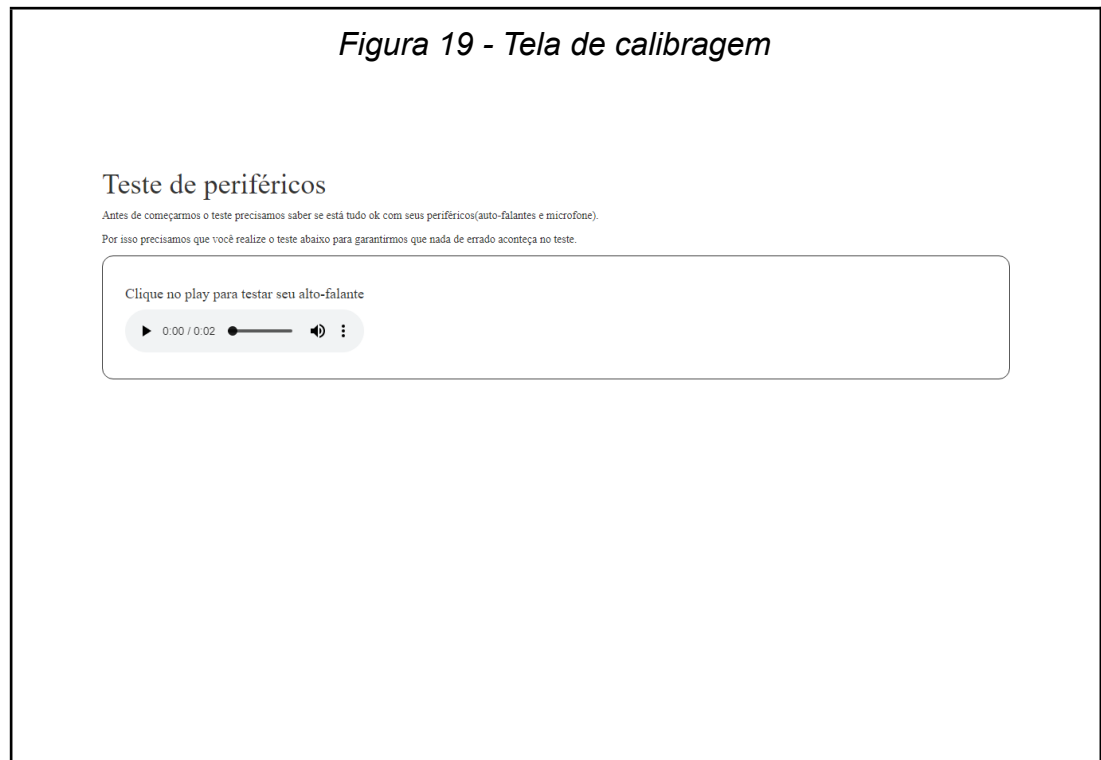
Descrição: Calibragem do sistema

Atores: Sujeito.

Pré-requisitos: UC 006

Tela:

Figura 19 - Tela de calibragem



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Usuário clica em “play”.
- 3 - Sistema emite a palavra de calibragem.
- 4 - Usuário confirma a escuta do som (E1).
- 5 - Sistema abre a opção de teste de de microfone.
- 6 - Usuário fala uma palavra.
- 7 - Sistema reconhece a palavra e habilita o botão “continuar” (E2).
- 8 - Usuário clica em continuar.
- 9 - Sistema carrega o UC 008.

Exceções:

- E1 - Usuário clica em “não escutei o som”
Sistema emite mensagem “aumente o som”
- E2 - Sistema não reconhece a fala do usuário
Sistema aguarda até o reconhecimento da fala do usuário

- UC 008: Atribuição de papel

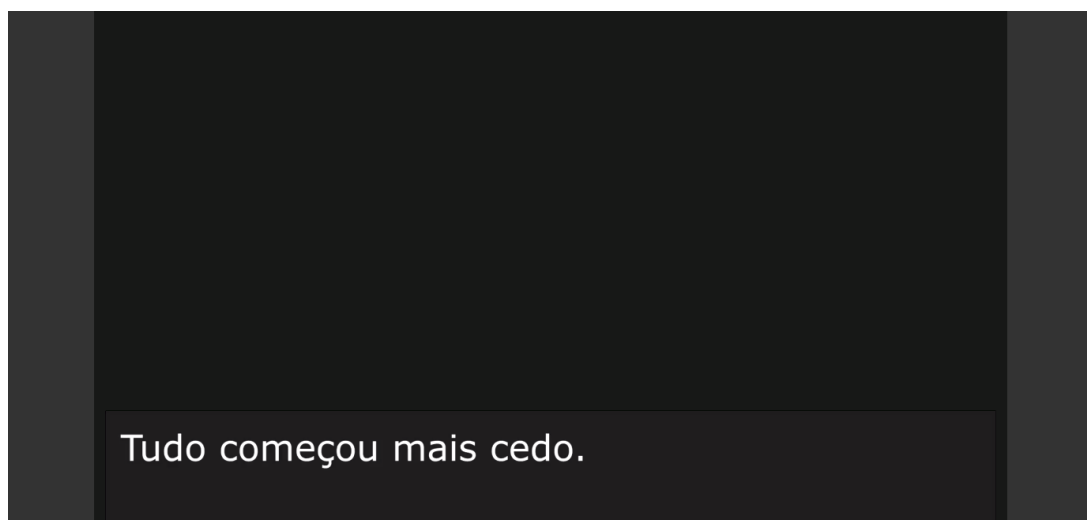
Descrição: Atribuição do papel ao sujeito

Atores: Sujeito.

Pré-requisitos: UC 007

Tela:

Figura 20 - Tela de atribuição de papel



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos.
- 2 - Sistema sorteia aleatoriamente um dos três papéis possíveis
- 3 - Sistema salva o papel sorteado
- 4 - Sistema carrega e inicia o video relacionado ao papel
- 5 - Ao final do video, sistema vai ao UC 009

- UC 009: Teste

Descrição: Atores: Sujeito.

Pré-requisitos: UC 008

Tela:

Figura 21 - Tela de teste



FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega todos os elementos do teste
- 2 - Sistema carrega uma palavra do teste e emite o som da palavra.
- 3 - Usuário emite sua resposta. (E1) (A1)(A2)
- 4 - Sistema armazena a resposta e tempo de reação do usuário.(A2)
- 5 - Sistema retorna ao passo 2 até serem respondidas todas as palavras
- 6 - Sistema emite mensagem de início da segunda fase
- 7 - Sistema repete os procedimentos de 2 a 5 até serem respondidas todas as palavras.
- 8 - Sistema verifica adequação das respostas.
- 9 - Sistema salva todos os dados.
- 10 - Sistema carrega UC 010: Resultados.

Fluxos alternativos:

A1 - Usuário fecha a página de teste.

1 - Teste é salvo no banco de dados como “incompleto”

Exceções:

E1 - Sistema reconhece uma palavra de forma incorreta ou não reconhece a palavra.

1 - Sistema segue com o procedimento após 5 segundos e anota uma “resposta incorreta”

- UC 010: Resultado

Descrição: Mostra o resultado da correção da rede neural

Atores: Sujeito.

Pré-requisitos: UC 009

Telas:

Figura 22 - Tela de resultado bom



Você consegue enganar o teste? Pressione F11 para sair do modo tela cheia

Você conseguiu enganar o teste e pode andar, em liberdade, em direção ao seu futuro!

FONTE: Os Autores (2018).

Figura 23 - Tela de ruim para o personagem Ladrão

Você não conseguiu enganar nosso teste

O teste acusou sua culpa no roubo do diamante e agora você passará um tempo na prisão.

Será que lá é tão ruim assim?

FONTE: Os Autores (2018).

Figura 24 - Tela de ruim para o personagem Testemunha

Você não conseguiu enganar nosso teste

O teste acusou sua presença no incidente e agora você será levado para um lugar desconhecido

O que será que vai acontecer agora?

FONTE: Os Autores (2018).

Figura 25 - Tela de ruim para o personagem Espião

Você não conseguiu enganar nosso teste

O teste acusou sua culpa no incidente e agora você será preso, pode tempo indeterminado para um interrogatório.

O que será da guerra enquanto isso?

FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega os resultados do teste e a rede neural
- 2 - Sistema corrige o teste com a rede neural e apresenta um resultado
- 3 - Sistema verifica se o resultado confere com o papel definido pelo sorteio de papéis
- 4 - Sistema mostra a tela adequada ao resultado obtido pelo usuário.

- UC 011: Login

Descrição: Login no sistema para visualizar os dados

Atores: Pesquisador.

Pré-requisitos: nenhum

Tela:

Figura 26 - Tela de Login

Usuário ou senha inválidos.

Acesso visualiação de dados:

Login

Senha

Acessar

FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega a tela
- 2 - Usuário digita o nome e a senha (E1)
- 3 - Sistema confere a senha(E2)
- 4 - Sistema encaminha o usuário para o UC 013

Exceções

E1 - Usuário não preenche todos os campos de senha

- 1 - Sistema emite mensagem solicitando preenchimento de todos os campos

E2 - Usuário preenche senha ou usuário incorretos

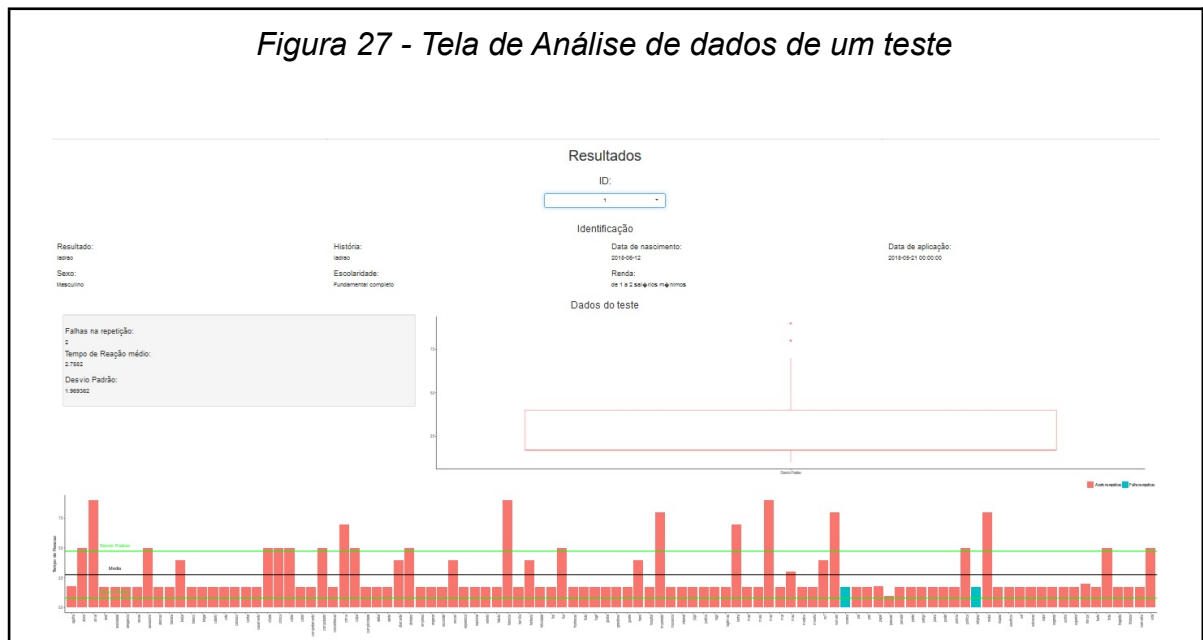
- 1- Sistema emite a mensagem de “senha” ou “usuário” incorreto.

- UC 012: Análise de Dados

Descrição: Mostra obtidos com os testes.

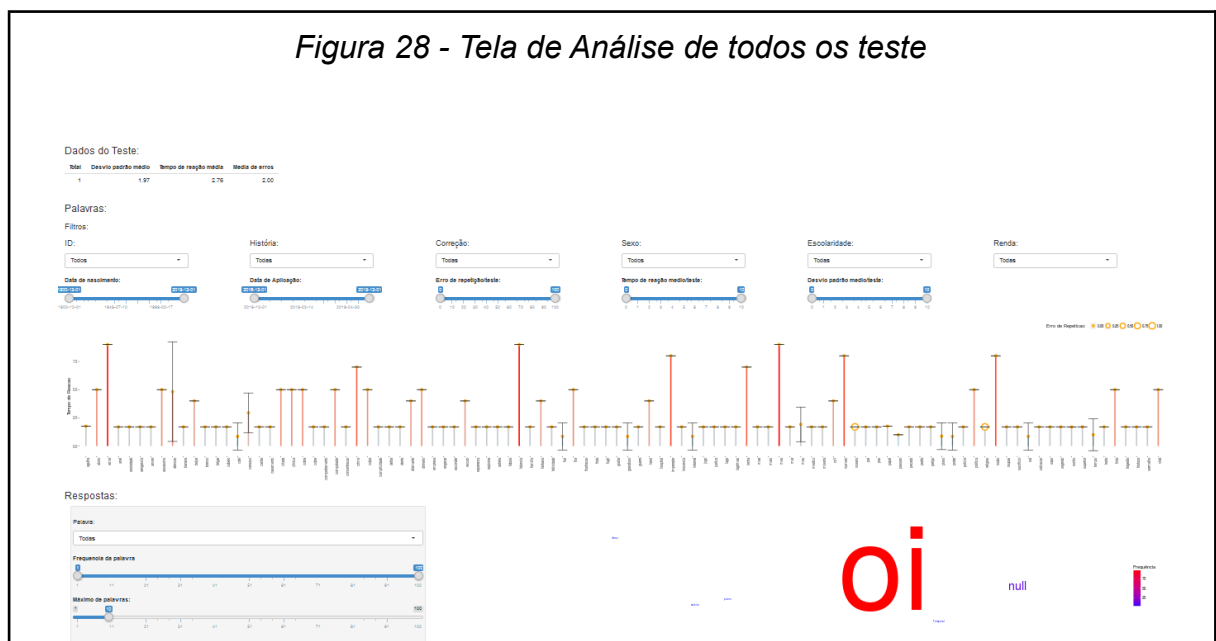
Atores: Pesquisador.
 Pré-requisitos: UC 011
 Tela:

Figura 27 - Tela de Análise de dados de um teste



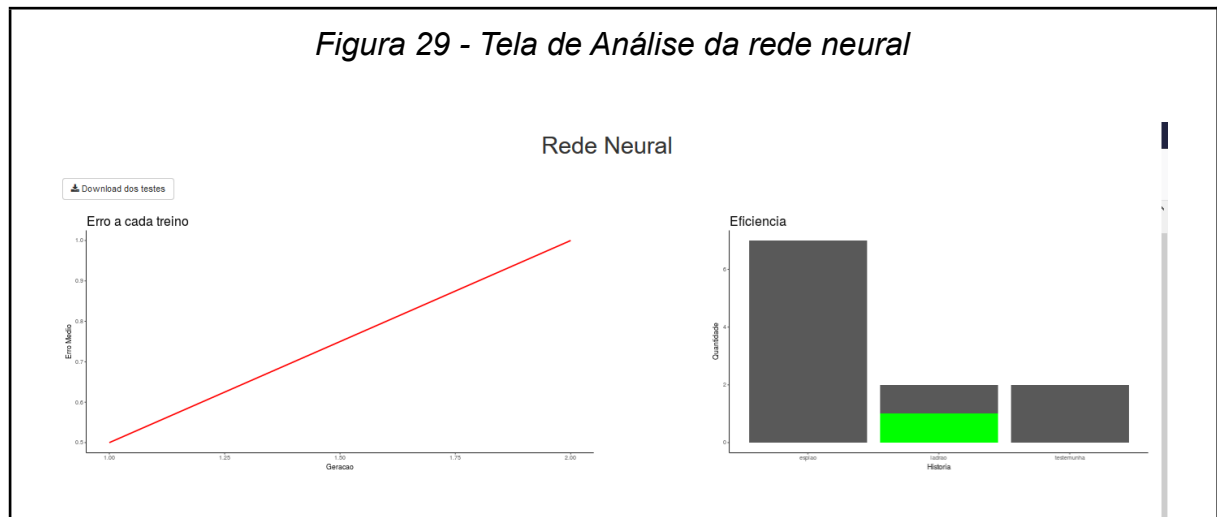
FONTE: Os Autores (2018).

Figura 28 - Tela de Análise de todos os teste



FONTE: Os Autores (2018).

Figura 29 - Tela de Análise da rede neural



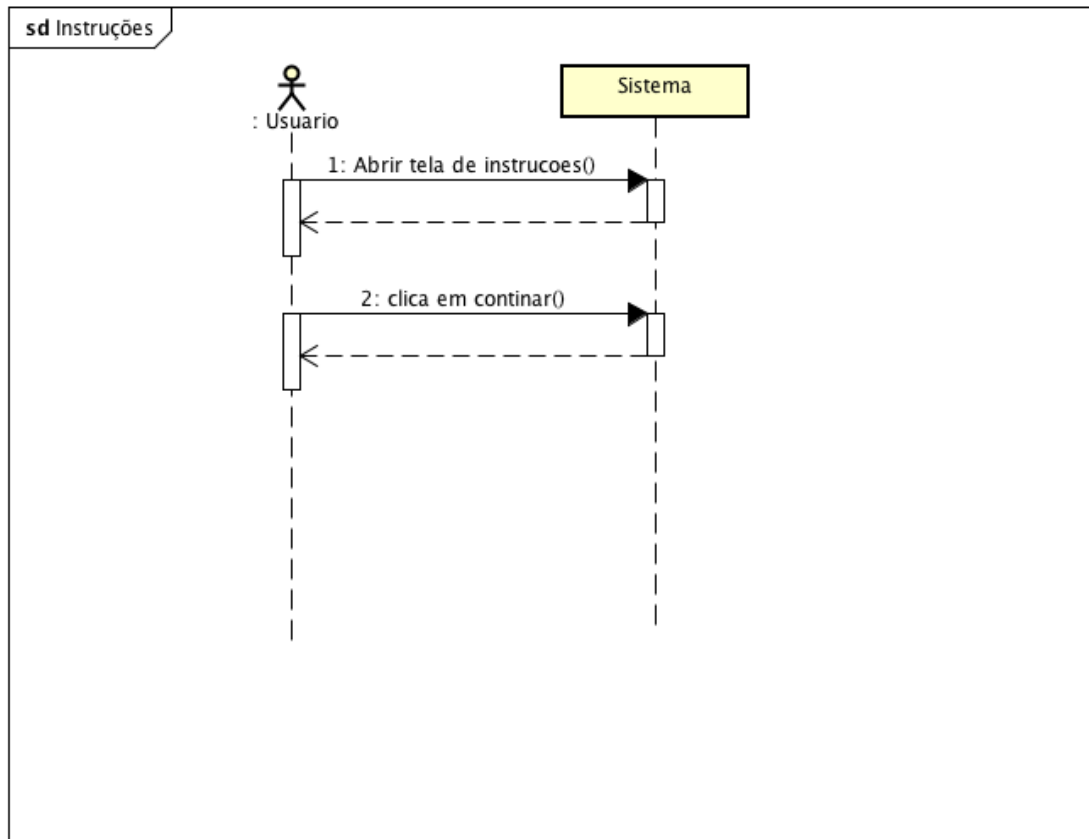
FONTE: Os Autores (2018).

Fluxo principal:

- 1 - Sistema carrega a página inicial de análise de dados
- 2 - Usuário seleciona quais dados deseja visualizar
- 3 - Sistema filtra os dados e mostra o que foi selecionado na tela.

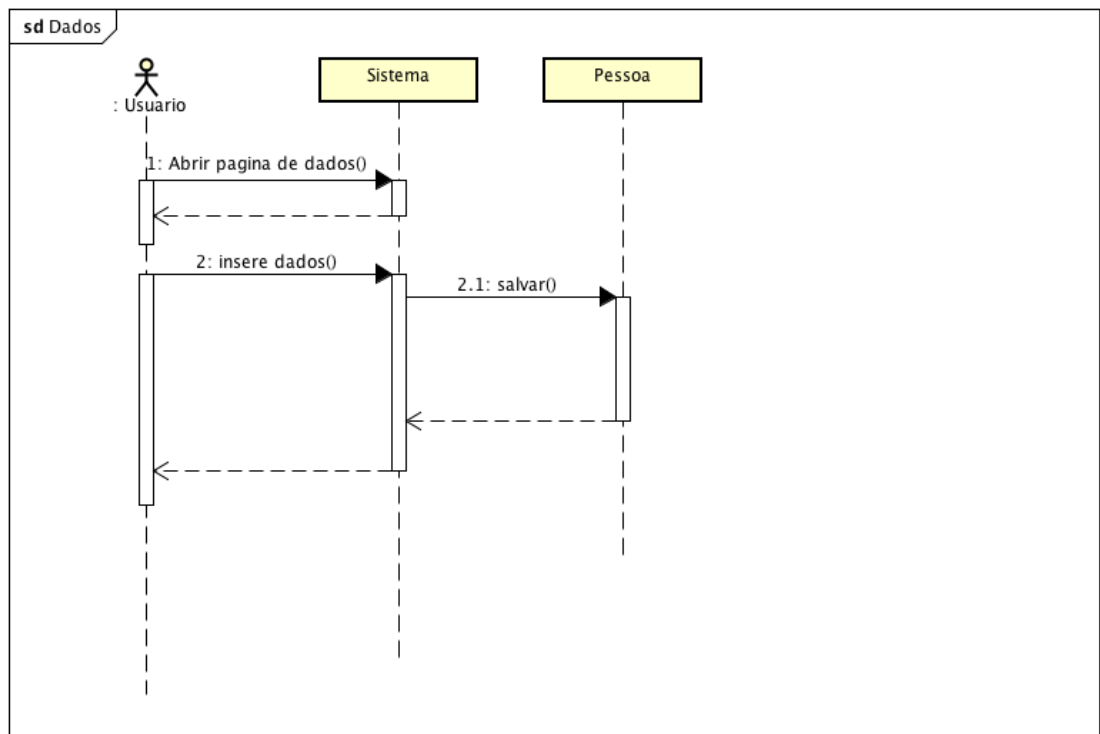
APÊNDICE D - DIAGRAMAS DE SEQUÊNCIA

Figura 30 - Sequência instruções



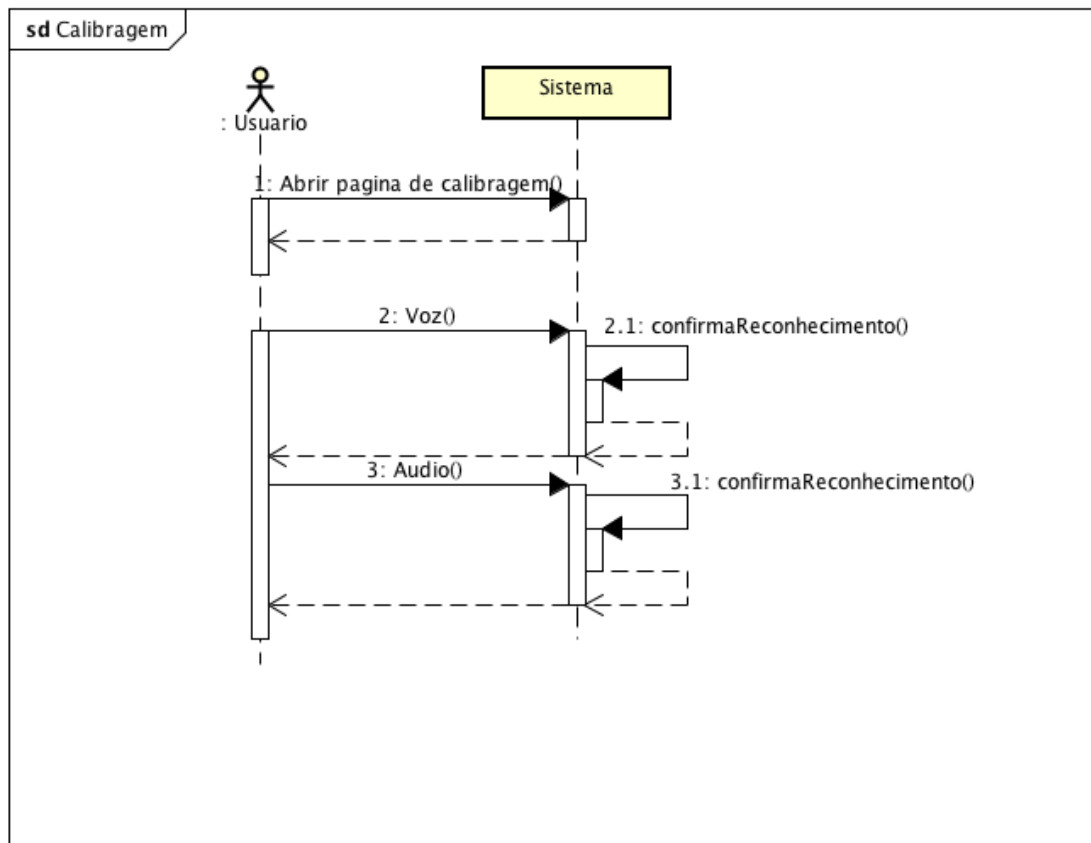
FONTE: Os Autores (2018).

Figura 31 - Sequência Dados



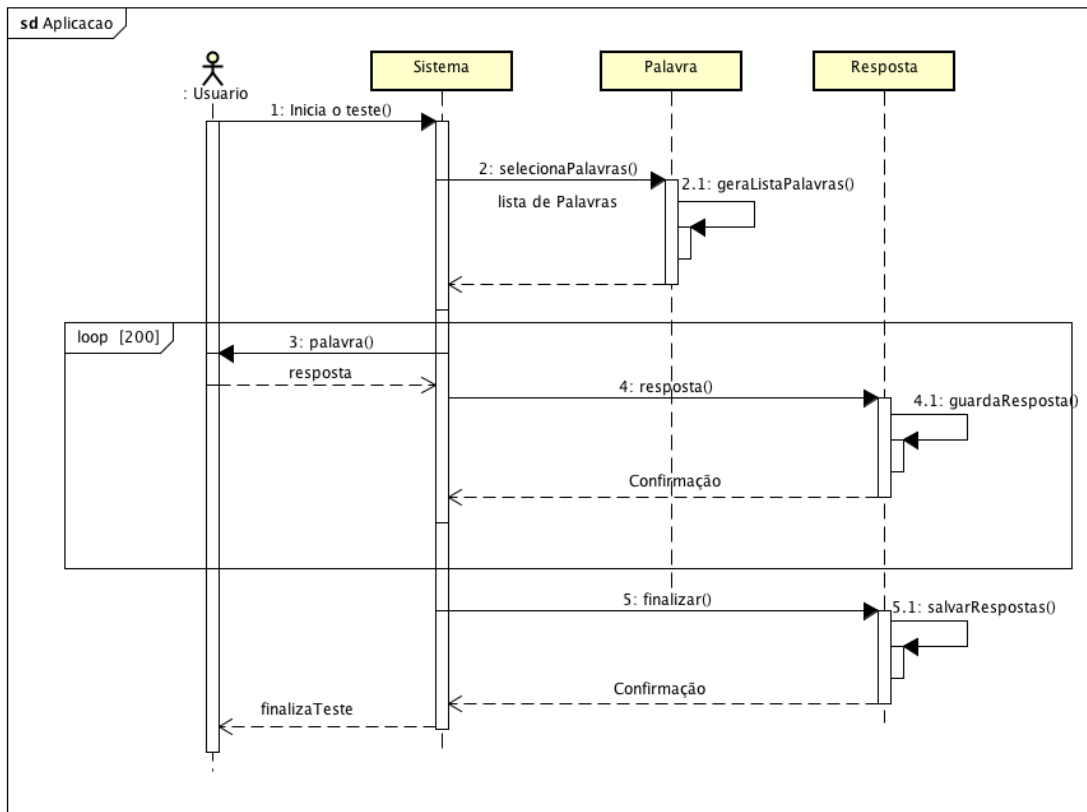
FONTE: Os Autores (2018).

Figura 32 - Sequência Calibragem



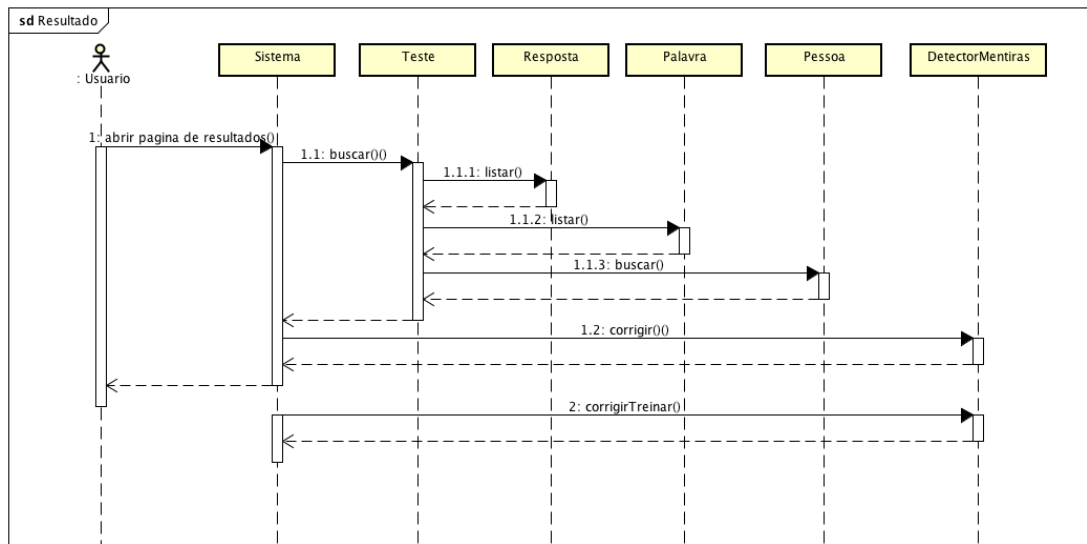
FONTE: Os Autores (2018).

Figura 33 - Sequência Aplicação



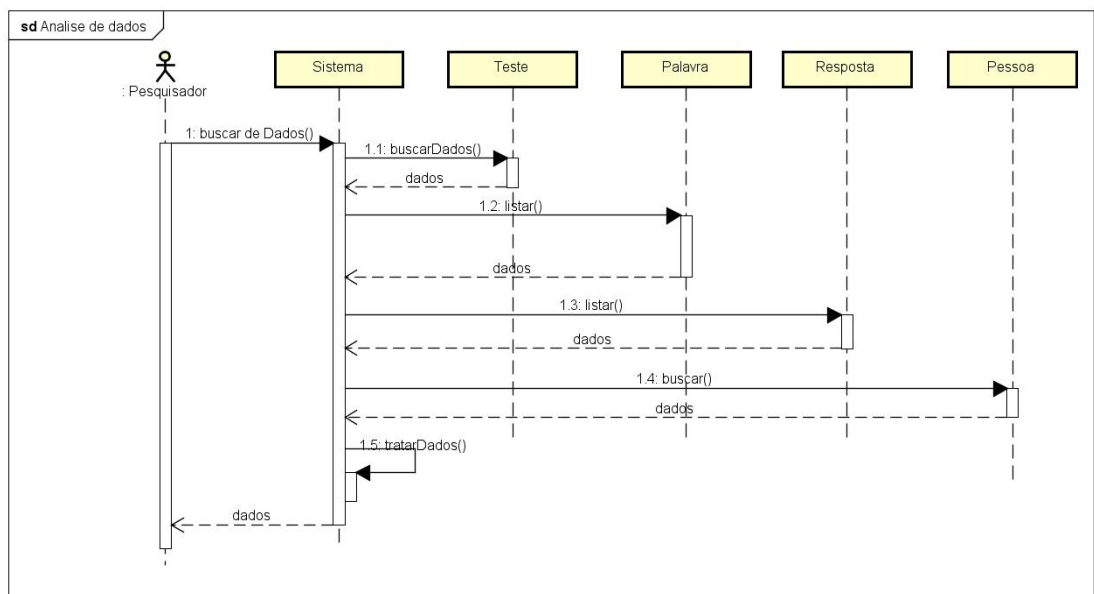
FONTE: Os Autores (2018).

Figura 34 - Sequência Resultado



FONTE: Os Autores (2018).

Figura 35 - Sequência Análise de dados

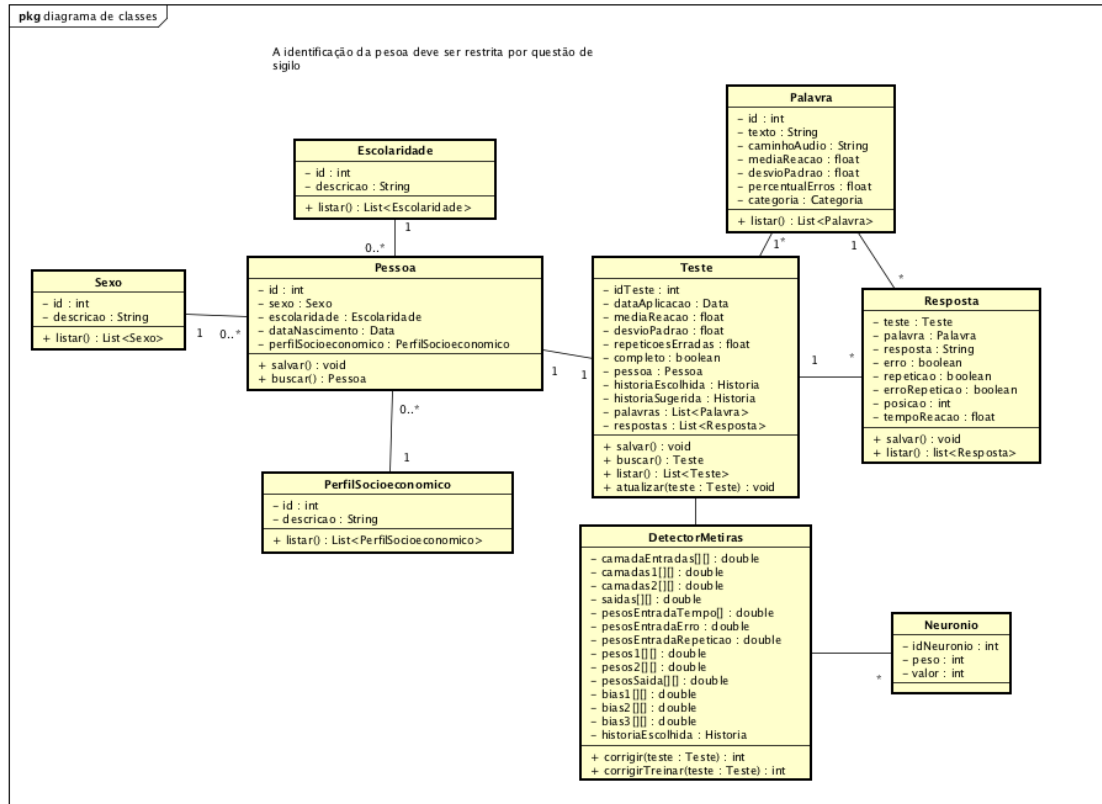


powered by Astah

FONTE: Os Autores (2018).

APÊNDICE E - DIAGRAMA DE CLASSES

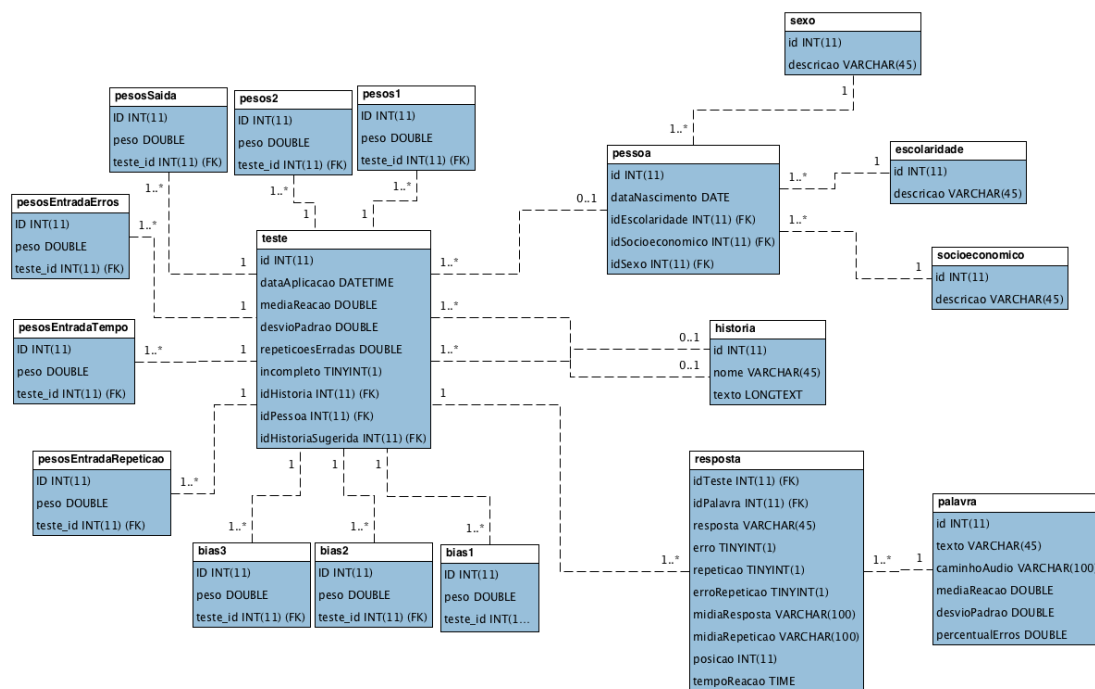
Figura 36 - Diagrama de classes



FONTE: Os Autores (2018).

APÊNDICE F - DIAGRAMA DE BANCO DE DADOS

Figura 37 - Diagrama físico de banco de dados



FONTE: Os Autores (2018).